

Worcester Polytechnic Institute Digital WPI

Masters Theses (All Theses, All Years)

Electronic Theses and Dissertations

2012-04-26

Predicting Network Performance for Internet Activities Using a Web Browser

Mihajlo Zeljkovic
Worcester Polytechnic Institute

Follow this and additional works at: <https://digitalcommons.wpi.edu/etd-theses>

Repository Citation

Zeljkovic, Mihajlo, "Predicting Network Performance for Internet Activities Using a Web Browser" (2012). *Masters Theses (All Theses, All Years)*. 335.
<https://digitalcommons.wpi.edu/etd-theses/335>

This thesis is brought to you for free and open access by [Digital WPI](#). It has been accepted for inclusion in Masters Theses (All Theses, All Years) by an authorized administrator of Digital WPI. For more information, please contact wpi-etd@wpi.edu.

Predicting Network Performance for Internet Activities Using a Web Browser

by

Mihajlo Zeljkovic

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

May 2012

APPROVED:

Professor Craig Wills, Thesis Advisor and Department Head

Professor Mark Claypool, Thesis Reader

ABSTRACT

Internet measurements have previously been done mostly from research labs and universities. The number of home users is growing rapidly and we need a good way to measure their network performance. This thesis focuses on building a web application that allows users to check what their network is like for online activities they are interested in. The application has minimal impediment to use by only requiring a Web browser. A list of online activities we offer users to choose from includes browsing web site categories such as news or social networks, having voice and video conferences, playing online games and other activities.

Acknowledgements

I would like to express my sincere gratitude to Professor Craig Wills for his invaluable guidance and support. He helped me focus on important aspects of the project and made completing this thesis possible.

I would also like to thank Professor Mark Claypool for reading my thesis and providing me advice on numerous occasions throughout the development of this work. My thanks are also due to my lab colleague Murad Kaplan who worked on the same project and was always ready to assist me when needed.

Table of Contents

1. Introduction	1
2. Background	3
2.1 Special Hardware Requirement	3
2.2 Special Software Requirement.....	3
2.3 Java Requirement	5
2.4 Browser Sandbox Methodology.....	5
2.5 Summary	6
3. Methods and Their Evaluation.....	7
3.1 Methods.....	7
3.2 Evaluation Setup	8
3.3 Download Throughput	9
3.4 Upload Throughput.....	10
3.5 Round-trip Time and Jitter	11
3.6 Measurements Considerations	13
3.7 Summary	14
4. Approach.....	15
4.1 Web Browser Basic Network Measurements	15
4.1.1 JavaScript DOM.....	16
4.1.2 JavaScript XMLHttpRequest	16
4.1.3. Methods Applicability to Different Servers.....	16
4.2 Derived Measurements	16
4.3 Ratings.....	17
4.4 Measurements Factors	17
4.5 Summary	18
5. Converting Network Performance to Activity Performance.....	20
5.1. VoIP and Video Conference	20
5.2 Online Games.....	22
5.3 File Hosting.....	23
5.4 Site Activities.....	24

5.5 Other Activities	25
5.6 Session History	25
5.7 Summary	26
6. Web Application.....	27
6.1 Design.....	27
6.2 Implementation	27
6.3 Database	32
6.4 Summary	33
7. Results.....	35
7.1 User Stats	35
7.2. Low-Level Results.....	36
7.3 Activities Popularity	38
7.4 User Activity Performance	40
7.5 Feedback and Additional Activities.....	50
8. Packet Pair.....	52
8.1 Calculations.....	52
8.2 Assumptions.....	52
8.3 Measurements	53
8.4 Web Site Results	58
8.5 Summary	60
9. Conclusions	62
10. Future Work	63
11. References	64

List of Figures

Figure 1 – Setup for Evaluation of Methods Used	8
Figure 2 – Download Results for Windows 7	9
Figure 3 – Upload results for Windows 7	11
Figure 4 – RTT CDF Results for Windows 7	12
Figure 5 – Site Home Page	28
Figure 6 – Users Feedback about Their Connection	29
Figure 7 – Explanation about Ratings	30
Figure 8 – Detailed Explanation about Ratings	31
Figure 9 – Session History of Ratings	31
Figure 10 – Users Feedback about Additional Activities and Feedback They Want to Share	32
Figure 11 – Rating Percentages for News Web Sites	41
Figure 12 – Rating Percentage for Social Network and Shopping Sites	42
Figure 13 – Rating Percentage for File Hosting Services	42
Figure 14 – Rating Percentage for Online Games	44
Figure 15 – Rating Percentage for Voice and Video Conference	45
Figure 16 – News Sites Rating Comparison for Home and Work/School Users	46
Figure 17 – Social Network and Online Shopping Sites Rating Comparison for Home and Work/School Users	47
Figure 18 – File Hosting Rating Comparison for for Home and Work/School Users	48
Figure 19 – WoW and LoL Rating Comparison for Home and Work/School Users	48
Figure 20 – First Person Games Rating Comparison for Home and Work/School Users	49
Figure 21 – Voice and Video Only Conference Rating Comparison for Home and Work/School Users	50
Figure 22 – CDF Graph for Packet Pair Measurements for NY Times Server from a Public Location	54
Figure 23 – CDF Graph for Packet Pair Measurements for BBC Server from a Public Location	55
Figure 24 – CDF Graph for Packet Pair Measurements for Facebook Server from a Public Location	56
Figure 25 – CDF Graph for Packet Pair Measurements for Ebay Server from a Public Location	57
Figure 26 – CDF Graph for Packet Pair Measurements for Origin Server from a Public Location	58

List of Tables

Table 1 – Measurements Used for Performance Estimation for IM, VoIP and Video Conference	20
Table 2 – Measurements Used for Performance Estimation for Online Games	22
Table 3 – Measurements Used for Performance Estimation for File Hosting	23
Table 4 – Measurements Used for Performance Estimation for Browsing Web Sites	24
Table 5 – Measurements Needed for Performance Estimation for Activities That Are Not Implemented	25
Table 6 – Number of Users in Different Groups.....	35
Table 7 – Low-Level Measurements Results for Different Locations	36
Table 8 – Low-Level Measurements Results for Different Types of Location	37
Table 9 – Low-Level Measurements Results for Different Types of Connection	37
Table 10 – Low-Level Measurements Results for Different Types of Internet Access	38
Table 11 – Percent of Users that Selected Each of the Activities	39
Table 12 – Percent of Users that Selected Each of the Additional Activities	51
Table 13 – Packet Pair Results for the Web Application Users at Different Locations.....	59
Table 14 – Packet Pair Results for the Web Application Users at Different Types of Locations	59
Table 15 – Packet Pair Results for the Web Application Users with Different Types of Connection.....	60
Table 16 – Packet Pair Results for the Web Application Users using Different Types of Internet Access	60

1. Introduction

Previously, Internet measurements have mostly been done from research labs and universities. The number of access points from residencies and public places has increased significantly. These access points are usually behind Network Address Translation (NAT) boxes and therefore users are not visible.

It is important to find new ways for measuring network performance of home users because they represent vantage points on the Internet that are not visible using other methods. We want to get as many users as possible to use our tool. We can achieve this by offering them some incentives. Previous research done in this area used methods such as providing users insight into their performance and how well they compare to other users.

Internet usage is constantly increasing. 77% of US adults are using it for reading email, playing online games, downloading music and videos, sending instant messages and many others online activities [1].

This thesis focuses on using a Web browser to estimate network performance for online activities users are interested in. We created a site where users can go to, select activities and get quick performance estimates. Our Web application has minimal impediments to use since it runs in a Web browser and has no additional software needed. It also has good incentives to attract users by giving them information about their network we believe is relevant to them.

Users care mostly about how effectively their network connection supports their online activities. In this application we provide them with information about activities such as browsing certain

types of websites, having a voice conference with a remote friend, playing online games etc. High level information about network performance for activities is important to them and can easily be understood by most users. They do not need to know the low-level details, just how their Internet habits are affected. We do however record low-level measurements such as download throughput and round-trip time. This data is not of particular interest to the users, but it is to the network researchers.

In order to use our tool, users only need to have JavaScript that exists in all modern Web browsers and is enabled by default. Each user is able to provide optional feedback that consists of free form input, selecting additional activities they want us to measure performance for, and multiple choice questions about their type of connection and location.

Chapter 2 describes work done in the area of home network measurements and Chapter 3 describes evaluation of the methods we use. Chapter 4 describes our approach to the home network measurements and Chapter 5 describes how we present the information we gather to users. Chapter 6 has details about the design and implementation of the website and Chapter 7 presents the results we gathered. In Chapter 8 we discuss the packet-pair technique and how we deployed it in our project. Chapter 9 derives conclusions from the results. In Chapter 10 we discuss improvements to our techniques that could be implemented in the future.

2. Background

This section provides previous work done in the area of home network measurements.

2.1 Special Hardware Requirement

The project in [2] obtains home network measurements by adding a router between modem and home network then performing both active and passive network measurements. Active measurements include throughput, latency, loss, jitter, DNS lookup and traffic shaping. Passive include wireless signals, access points and aggregated traffic information. The router sends a UDP packet back to the project host server every two minutes.

Currently, there are 16 routers deployed. Most of them are in Atlanta area and they all perform measurements to Georgia Tech. This approach is gathering more thorough information about network than ours, but it has larger impediments to use because users have to order and install the router. As an incentive, users can access their low-level measurements. In our Web application, users can view the low-level measurements, but they can also get information about the effect of these measurements on their online activities.

2.2 Special Software Requirement

NETI@home [3], DIMES [4] and DipZoom [5] request a user to install their program and therefore have high impediments.

NETI@home uses a C++ program built on top of network sniffer. The program analyzes the data and sends a summary to the project base. It focuses on the following four transport-layer protocols: TCP, UDP, ICMP and IGMP.

DIMES [4] studies structure and topology of the Internet. The project focuses on user location and performs measurements such as *traceroute* and *ping*. Users have the ability to see where the nodes they are communicating with are located.

DipZoom [5] allows users to offer and request measurements between their computing devices. Measurements include *ping* for measuring RTT to the target, *traceroute* for obtaining path, *wget* for downloading any Web page and *nslookup* for measuring DNS query delay.

Projects mentioned previously do not require any feedback from users before or after performing measurements. The approach in [6] uses software to analyze irritation of users caused by different network parameters allowing them to indicate when they are dissatisfied with how their network is performing. Software is able to inspect network packets and gather data such as throughput, RTT, link properties and wireless signal strength. It can also collect data about local machine: applications that are currently used, application with a window focused, CPU utilization and mouse and keyboard usage.

The study was performed over three-week period and included 32 participants. Most of them were non-tech users from Northwestern University students, faculty and staff. Each of them had a sticker on the laptop that reminded them which button to press if they are irritated by the network. Results of the study showed that users can decide when network is responsible for their irritation, and that their irritation is affected by application and their location.

Projects in this section provide less measurements and incentives for home users to participate than the tool we use. All measurements are done at the low level and applications used are only considered in [6].

2.3 Java Requirement

The Netalyzr project [7] analyzes Internet connection properties such as latency, bandwidth, blocking of services, HTTP caching, proxy correctness, NAT detection and user DNS server's resilience to abuse. This data is gathered using Java Runtime Environment (JRE) and therefore imposes limited impediment such as installing it and giving access permission. Users have the ability to see the results and provide feedback about their type of location and access point using an optional questionnaire. JRE is unable to detect this information about users due to its limitations.

The project in [8] uses a signed Java applet and also has limited impediments for participation. It gathers information about the configuration of the user's machine, wireless connectivity, download and upload throughput, DNS performance and the number of devices connected to the same network.

Installing JRE and giving permission to use has larger impediments than the tool we use. Projects in this section gather low-level measurements that are impossible for us to infer within a Web browser, but none of these measurements are what a typical user would be interested in which provides little incentive for them to use tools like this.

2.4 Browser Sandbox Methodology

SpeedTest [9] runs in a Web browser and has minimal impediments to use by requiring only a Web browser and Adobe Flash. It allows users to measure their round-trip time and download and upload throughput to any of their servers. This approach gives users some incentives and the ability to check if their Internet provider is giving them the promised bandwidth, but it does not

give them the high-level picture of what those results mean in terms of affecting their online activities.

The project in [10] performs network measurements in a Web browser using JavaScript and Adobe Flash. We are using some of the same methods for this thesis.

2.5 Summary

Previous work done in this area has impediments for use. These impediments included installing additional software or even setting up an external hardware such as a router. Users were offered limited incentives such as viewing low-level measurements that they or other participants obtained. We use a tool that has minimal impediments to use by requiring a Web browser only. It also provides good incentives to typical users by displaying information about network effect on their online activities.

3. Methods and Their Evaluation

3.1 Methods

This thesis work is based on Sandbox methodology from [10] to estimate file download and upload time. Methods used in the project are: JavaScript DOM, JavaScript XMLHttpRequest (XHR), Flash load-policy file (LPF) and Flash URL.

XHR and URL methods measure time that takes to download or upload a file to the origin server using JavaScript or Flash. Dividing file size with this time gives us download and upload throughput. If we download a file that fits into one TCP/IP packet over an established TCP connection, we get round-trip time (RTT).

LPF retrieves policy file that lists all domains that are allowed to access. If the origin server is on the list, the file is retrieved, otherwise an exception is thrown. This method can be used to obtain RTT for arbitrary server by measuring how long it takes to catch an exception and it can be used for download throughput for origin server and any other server that has our domain in their policy file.

DOM measures the time that takes to download a file from arbitrary server and put it in an `<iframe>` tag. If the file is large, we can obtain download throughput. If it fits into one TCP/IP packet, we can obtain RTT.

All of these methods are used for measuring the time it takes to perform download and upload of a file at the network level. Performing measurements at the application level introduces overhead that was not well understood. The overhead was evaluated using controlled network

environment. We describe an overview of the results in this chapter. More details are available in [11].

3.2 Evaluation Setup

Figure 1 shows the setup used for evaluation. Measurements were done on two computers, new desktop and older laptop using Windows 7 and Ubuntu Linux. Network traffic was controlled using a bridge with two network cards, one for incoming and the other one for outgoing traffic. Download throughput was 1Mbps and upload throughput was 0.25 Mbps. Both cards had a latency of 50 milliseconds, resulting in round-trip time of 100 milliseconds.

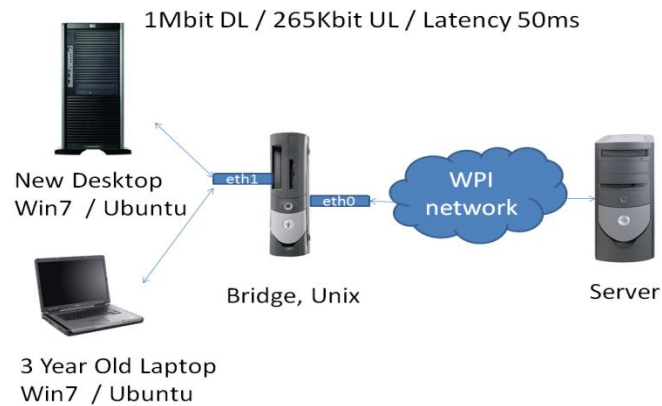


Figure 1 – Setup for Evaluation of Methods Used

Browsers used for these measurements included Chrome, Firefox and Internet Explorer. As ground truth measurements, a compiled C program executing GET and POST requests was used. For download and upload tests, a range of object sizes from 1KB to 1MB was used. Each file was downloaded or uploaded 10 times using each of the four techniques.

3.3 Download Throughput

Figure 2 shows median download throughput for each of the 10 measurements for each method for Windows 7 Chrome and Internet Explorer.

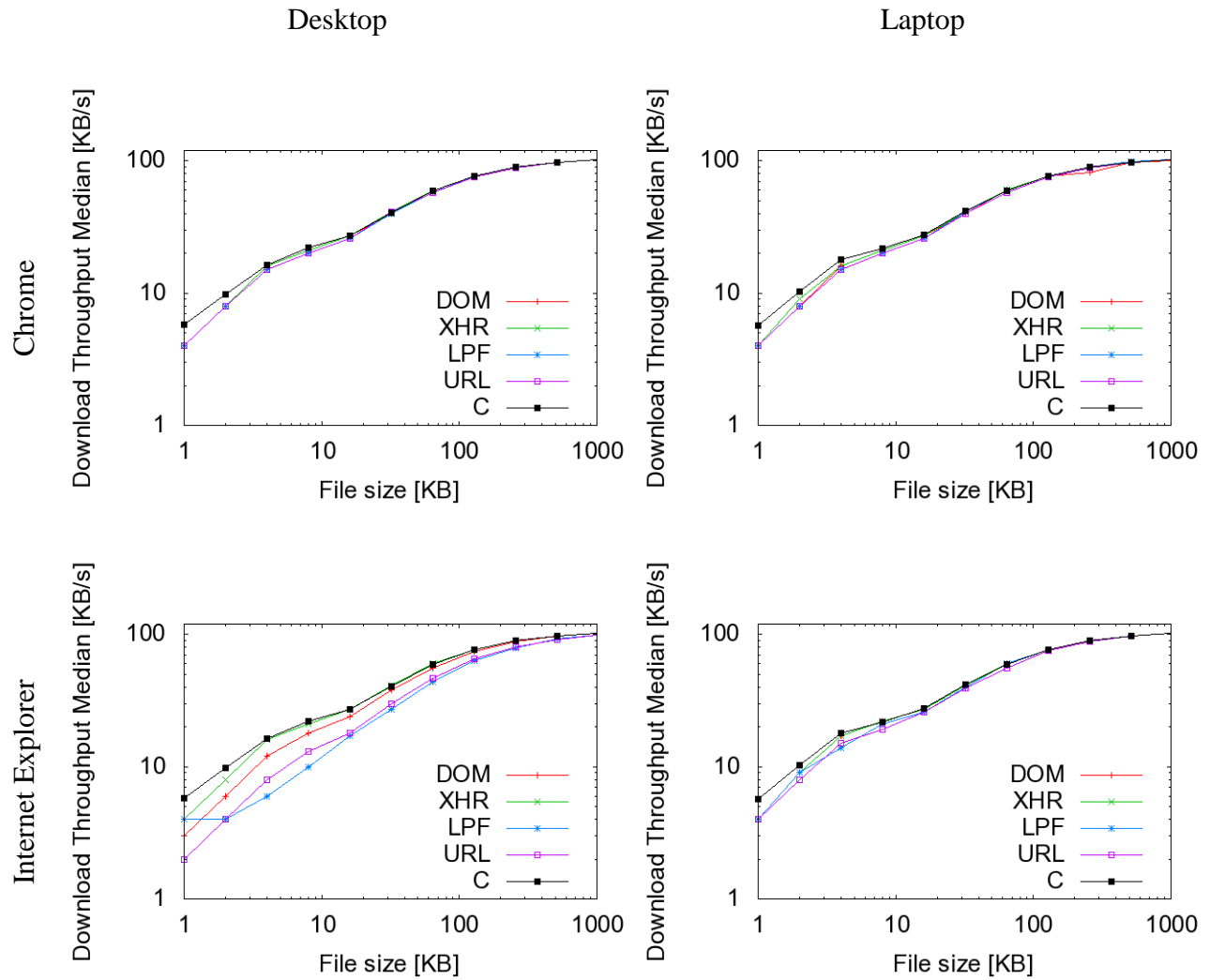


Figure 2 – Download Results for Windows 7

Results obtained show that we can accurately determine download throughput using JavaScript methods as long as the file is being downloaded for at least a couple of seconds. Flash methods have more overhead and they require longer downloads for good estimate.

3.4 Upload Throughput

Figure 3 shows median upload throughput for each of the 10 measurements for each method for Windows 7 Chrome and Internet Explorer.

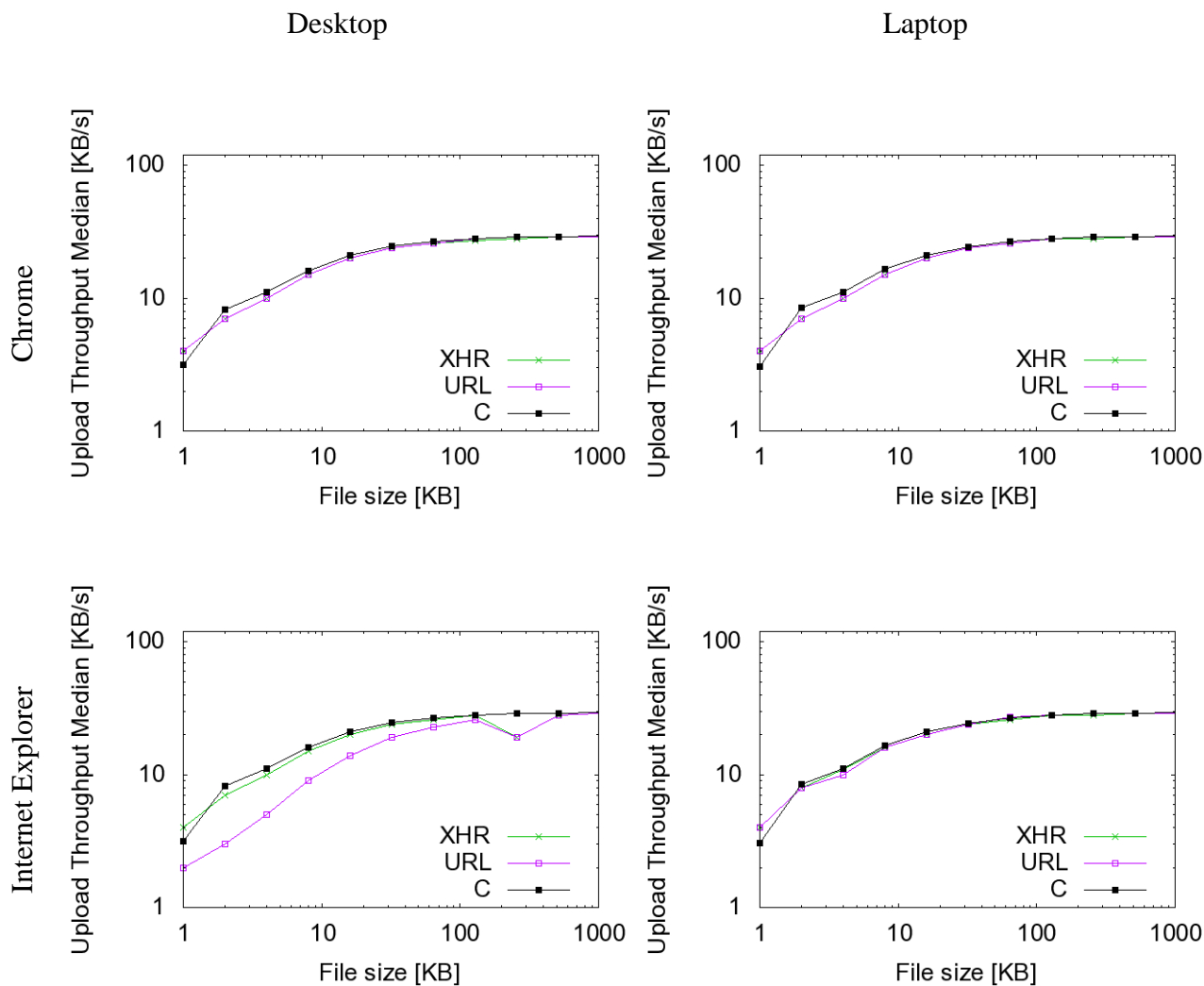


Figure 3 – Upload results for Windows 7

Results obtained show that we can use the XHR method to estimate upload throughput as long as we upload data for at least couple of seconds. The URL method has more overhead and it requires longer upload for a good estimate. We test only XHR and URL methods because DOM and LPF cannot be used for upload measurements.

3.5 Round-trip Time and Jitter

For round-trip time and jitter measurements, 50 consecutive round-trip time measurements were done. CDF graphs for each method for Windows 7 Chrome and Internet Explorer are shown in Figure 4.

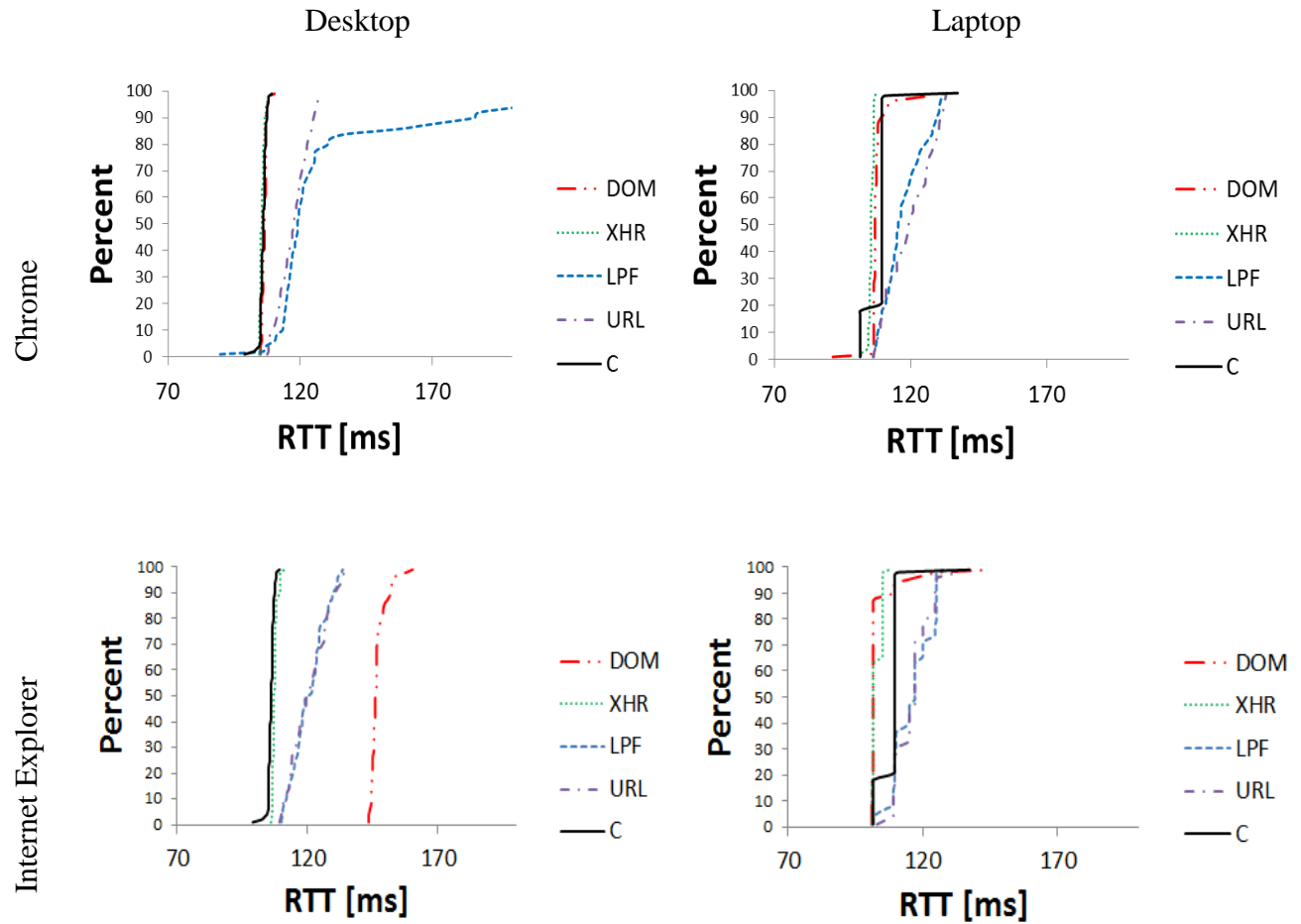


Figure 4 – RTT CDF Results for Windows 7

Results obtained show that JavaScript methods can be used effectively for measuring round-trip time and jitter. Internet Explorer results on one of the computers used showed consistent results for DOM method, but are off by 50 milliseconds from the ground truth. In the work done for this thesis, we compensate for overhead in each of the methods used by deducting time that takes to download an object from the cache. We assume that the browser overhead can be estimated by taking the time it takes a user to download an object that is located at the same machine.

3.6 Measurements Considerations

We need to consider factors that could affect the low-level measurements we obtain in the Web browser:

1. Browser overhead

As shown in [11] there is a significant overhead of the methods used. This overhead can increase the time we measure file download or upload. The effect is greater for smaller times, such as the ones for RTTs. To deal with this issue, we deducted the time it takes browser to retrieve an object from cache. Our assumption is that retrieving an object that is already located on the machine is similar to the browser overhead.

2. DNS query and connection establishment

DNS lookup could add to the RTT measurements. For throughput measurement using large files, this is negligible. For RTT measurements, it could have a significant effect. This is why we execute the first measurement, but we do not use it. First measurement is also used to establish the connection and we assume that every request we make within less than one second after the previous one will have a connection already established.

3. Server processing delay

A HTTP server requires a certain amount of time to parse and process request and return the requested response. This could be a non-negligible amount of time, but we do not take it into account.

4. Maximum packet size

We expect that the maximum size of TCP/IP packet is 1500 bytes. If the data we requested and the header attached by the server do not exceed this limit, we expect to have one packet.

5. HTTP header size

When measuring throughput, we only consider size of the data transferred. We do not consider a few hundred bytes of header that are attached. This gives us lower throughput, but if the file used is large enough, error is negligible. This could present potential issue for RTT measurements, but we make sure that the header attached does not make the response exceed 1500 bytes and that it fits a single packet.

3.7 Summary

We evaluated the Sandbox methodology from [10] using controlled network environment. Results showed that we can use these methods reliably to estimate round-trip time and download and upload throughput to arbitrary servers.

4. Approach

Our goal is to get information about users online activities and their network performance. For this purpose we made a Web site where users can go to and choose activities they are interested in. After choosing, we perform network measurements to predict performance and provide them explanations about it.

We use this approach in order to minimize impediments a user has to overcome to perform measurements. The only requirement is having a Web browser with JavaScript enabled. We also give users some incentives by showing them how these measurements affect the performance of their online activities. We believe that this information is more meaningful to typical users than just giving them raw measurements.

4.1 Web Browser Basic Network Measurements

There are three basic network measurements we can obtain within the Web browser: round-trip time, download throughput and upload throughput. Download throughput is obtained by dividing size of a file and the time it takes to download it. Upload throughput is calculated by dividing the size of data uploaded and the time elapsed. Round-trip time is measured as the time that takes to download a file that fits one TCP/IP packet. For performing download and upload requests, we use JavaScript DOM and JavaScript XHR methods because of their consistency with ground truth results in [11].

4.1.1 JavaScript DOM

We load an object from arbitrary server into an iframe tag on web page. We record start time when we change the source of the iframe and record end time when loading is done and “onload” event is executed.

4.1.2 JavaScript XMLHttpRequest

We download an object from the origin server and record the duration of the download process. We record start time when we send the request and end time when the request changes status to done. We can also record the time it takes us to upload a file to the origin server.

4.1.3. Methods Applicability to Different Servers

Both JavaScript methods can be used to obtain RTT and download throughput. XHR performs measurements only for the origin server. If we need to do perform measurements for other servers, we use DOM.

4.2 Derived Measurements

We have three additional measurements derived from the basic ones: loss, jitter and download/upload ratio. For jitter we need to obtain a set of consecutive RTTs executed with a pause between, and measure the variance.

For loss we also need a set of consecutive RTTs. We cannot directly measure TCP loss because of the sandbox restrictions, but based on our testing it can be estimated well as a percent of consecutive RTTs that are at least three times higher than the median. If TCP incurs loss, we can either get a triple duplicate ACK or a timeout. Since the response has only one packet, we cannot get a triple duplicate ACK. TCP sets timeout interval to be twice the estimated RTT, so this gives us a total of three RTTs. We do not account for the possibility that the loss occurred more

than once during RTT measurement. We also do not take into account that a larger RTT could be due to the congestion instead of loss.

Using XHR method we can get a download/upload ratio for hosting server and use it to estimate upload throughput to an arbitrary location. We assume that this ratio is going to be constant for any location. Dividing download throughput to an arbitrary location with the obtained ratio gives us the corresponding upload throughput.

4.3 Ratings

We use basic and derived network measurements to come up with a rating value for each activity. We present ratings as a score on a five-star scale. We chose to represent it this way because we believe this is the information that any user can understand easily. This type of scale is used on many websites and users are familiar with it. We also considered using written explanations for each rating, but ultimately decided to present it as a number of stars because we were concerned that words would be interpreted differently by different users.

4.4 Measurements Factors

Here are some of the important factors that might affect measurements in our application and how we handle each:

1. HTTP Persistent connection

Before deploying the application, we made sure that all servers we are using have persistent connections, and we assume this does not change.

2. Network congestion

Network failures and other hosts on the user's local network could have an effect on our measurements. However, any of these factors would also have an effect on user's performance in general too.

3. User Activity

If a user has any other activity that involves network traffic, performance we measure could be affected. In this case, RTT can be larger and throughput can be disabled from reaching its maximum because our application has to share bandwidth. However, we are still measuring the network under current conditions.

4. Constant ratio between upload and download throughput

We cannot get upload throughput to a third-party server directly using our methods. We can estimate it using download throughput to that server and upload and download throughput to the origin server. Our assumption is that the ratio between upload and download throughput to any server is constant.

5. Caching

In order to make Web browser download file each time instead of retrieving it from the cache, we append parameter representing number of milliseconds since the epoch. If we are downloading multiple files at once, like in Web site activities or packet pair, we also attach a unique iframe number.

4.5 Summary

In order to gather network measurements from as many typical users as possible, we need a tool that has minimal impediments to use. For this purpose, we made a Web site where users can go to and perform measurements for activities they are interested in. Using JavaScript, we perform

basic measurements such as round-trip time or download and upload throughput and from these we derive loss, jitter and download/upload ratio.

5. Converting Network Performance to Activity Performance

While raw values of our measurements might be interesting to the network researchers, a typical user may not be concerned with these. In order to provide them with the information that is more relevant to them and easier to understand, we map these values to five-star rating and give users some insight into the way we performed measurements and mapped them to expected activity performance. We give them two levels of explanations. The first includes measurements such as estimated time to download a page. It is written so that any user can understand it. The second explanation is more detailed and contains information that benefits users who are interested in details of network measurements.

We estimate network performance for Internet activities using the basic measurements gathered within the Browser sandbox environment. The list of activities was obtained from a survey conducted in the US that researched what the popular Internet activities are and how many people are using them [1].

5.1. VoIP and Video Conference

Table 1 – Measurements Used for Performance Estimation for IM, VoIP and Video Conference

Online Activity	Measurement				
	Download	Upload	RTT	Loss	Jitter
VoIP			✓	✓	
Video Only Conference	✓	✓			

Table 1 contains measurements we used for voice conference (VoIP) and video only conference. We wanted to clearly distinguish between the two, so VoIP does not include video and video only conference does not include voice.

Measurements are all performed to the location users indicate they want to communicate with. We estimate RTT to different servers in this area using the JavaScript DOM method. For each of them we measure consecutive RTTs certain number of times. Currently this number is set to be 20. It is large enough to give us good measurements and also small enough that a user does not wait for too long.

For instant messaging, we calculate the median of all measurements for each of the servers and take median of them as estimated RTT. We have predefined intervals of RTT for our five-star ratings and determine performance based on the estimated value.

For voice conference we use average RTT and loss rate to calculate E-model Mean Opinion Score (MOS) [12] for each server and take the median. Quality of voice conference is dependent on jitter too, but E-model does not directly consider it. It is only included as an increase in average RTT. We connect MOS with user satisfaction by mapping ranges from [13] to our five-star scale.

For video conference we take the results we had for voice conference and combine them with download and upload throughput measured to the desired location. Download throughput can be measured directly to the third party server and upload can be estimated using download/upload ratio to our server. Both download and upload need to be above certain threshold in order to get higher rating. Thresholds were obtained from the Skype minimum and recommended speeds for different video qualities [14].

5.2 Online Games

For online games performance, we ask users to choose the server they are using from the list of well-known public servers. If the game has private servers, we find closest servers for that particular game. We do this by screen scraping Game Tracker periodically to obtain the list of all game servers and combine this information with MaxMind Geo IP technology to get the ones that are closest [15] [16].

Table 2 – Measurements Used for Performance Estimation for Online Games

Online Activity	Measurement				
	Download	Upload	RTT	Loss	Jitter
Online Games - First person			✓		
Online Games - Third person			✓		
Online Games - Omnipresent			✓		

Table 2 shows the measurements we use for games network performance. For each area, we measure multiple consecutive RTTs to a server in it and calculate the average. We currently do 20 RTT measurements to each server. This is the number we believe is large enough to obtain a good set of measurements and also small enough that a user does not have to wait for too long. Online game performance is dependent on the jitter too, but we do not take it into account directly. It is only included as the increase in average RTT.

For first-person shooter, third-person games and omnipresent games we use results from [17] and create intervals using RTTs that decrease users performance to 60, 70, 80 and 90 percent as limits. We map average RTT to our five-star scale using these intervals.

We assume that most game servers have Web servers. Our crawler checks in advance if each game server has a Web server running on port 80. We assume that most of them have it installed

and therefore the closest game server will approximately be the closest game server we have access to.

We also assume that MaxMind coordinates are accurate enough. We use latitude and longitude for IP addresses to calculate the distance between them. If the coordinates provided are not accurate enough, we could miss some of the closest servers. According to the MaxMind web site, it is 99.5% accurate on a country level and 79% on a city level for the US within 25 mile radius [18].

5.3 File Hosting

Table 3 – Measurements Used for Performance Estimation for File Hosting

Online Activity	Measurement				
	Download	Upload	RTT	Loss	Jitter
File Hosting Sites	✓	✓			

Table 3 shows measurements we used for estimating performance for file hosting sites. We need download and upload throughput, and we measure them for the server that is close.

We use the closest Akamai server [19] for this purpose. Akamai’s distributed content delivery system has a lot of servers around the world. Many websites use their services to deliver content to users faster. Files that are often accessed get cached. We can use these files to get a good estimate of the download throughput to a close server.

We cannot measure upload throughput to an arbitrary server directly. We estimate it by dividing this speed by the download/upload ratio for the origin server. Both download and upload throughput need to be above certain threshold to get higher rating. We came up with these

thresholds by taking into account how quickly users can download larger files that are hosted on file hosting services.

5.4 Site Activities

Table 4 – Measurements Used for Performance Estimation for Browsing Web Sites

Online Activity	Measurement				
	Download	Upload	RTT	Loss	Jitter
News	✓				
Shopping	✓				
Social Networks	✓				

Table 4 shows measurements we use for estimating performance for browsing different categories of websites. We will present an overview of how we determine user ratings here. More information is available in the Master thesis done by Murad Kaplan at WPI [20] and a paper based on this thesis [21].

For each site of interest we download multiple objects from the servers providing the content and estimate download time of the main page as described in. Download time is calculated as:

$$t_l = t_c + \frac{t_a * n * p}{6}$$

t_l – time to load main page

t_c – time to load main page container

t_a – time to load 6 average size objects from the page concurrently

n – total number of objects on the page

p – portion of the objects coming from the dominant server

Works in [22] and [23] describe user satisfaction dependence on the page load time. These descriptions were used for mapping estimated page download times with five-star ratings.

5.5 Other Activities

Table 5 – Measurements Needed for Performance Estimation for Activities That Are Not Implemented

Online Activity	Measurement				
	Download	Upload	RTT	Loss	Jitter
Music Sharing	✓				
Video Sharing	✓				
Live Streaming Video	✓		✓		✓
Live Streaming Audio	✓				

Table 5 shows measurements needed for estimating performance for activities we currently do not have implemented because we do not have a method to provide feedback to users. We cannot get measurements directly from Web sites of interest and for some of the activities we cannot map them to the rating.

5.6 Session History

Session is a time period that starts from the moment users enter Web application to the moment they close it. Throughout the session, we periodically repeat all measurements a user selected. More information will give us reliable measurements and a better insight into how activity performance and low-level measurements change over time.

We can also show users rating results for entire session for each activity they selected. This will enable them to leave the application running in the background and see the history at any time.

5.7 Summary

We perform network measurements for some of the popular online activities such as browsing Web sites, VoIP or video conferencing, playing online games and using file hosting services. For each of these we present five-star scale rating and two levels of explanation. The first level includes values we used to get the rating and the second level includes the low-level measurements obtained.

6. Web Application

6.1 Design

We wanted to give users an idea of what the project is about on our main page and let them go to the separate tests page from there. The tests page includes a list of online activities and users could choose to perform measurements for any of them. Measurements are executed periodically if the application is left open.

There is an optional survey about users network details such as whether they are accessing Internet wired or wirelessly. These are the information that we are unable to infer due to the Sandbox limitations and therefore need users to provide them themselves. We also wanted to allow users to send comments about what they think is relevant and will help us improve the site.

We have three different levels of explanations. First is a five-star rating, second provides more insight about the reason for rating and third gives the low-level measurements used. We also wanted to allow users to track how they performed over time.

6.2 Implementation

We separated different parts of our application into internal tabs. We have tabs for front page, methodology, tests, history of results for all users, and history of results for users with similar characteristics. Each user is identified by IP address and cookie.

The application is hosted at hmn.cs.wpi.edu and its front page is shown in Figure 5. It has a brief explanation about the project and a link to tests page.

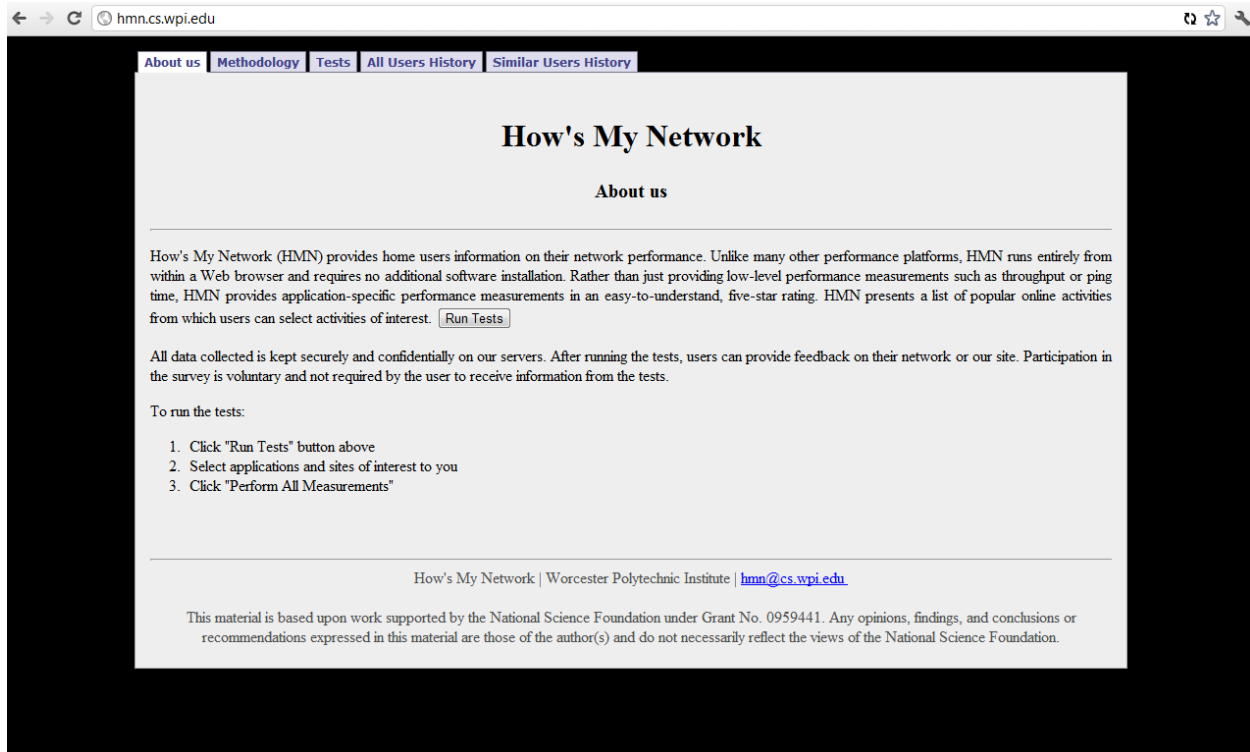


Figure 5 – Site Home Page

On the tests page, users are able to provide information about their connection. As shown in Figure 6, this information includes the type of connection and type and location of access.

← → ↻ hmn.cs.wpi.edu

[About us](#)
[Methodology](#)
[Tests](#)
[All Users History](#)
[Similar Users History](#)

How's My Network

Performance Tests

Filling in your connection settings will let you compare your results to users similar to you:

What type of Internet connection do you have? ☐ Dial-up ☐ DSL ☐ Cable ☐ Fiber ☒ Business ☐ Other ☐ Don't know

How are you accessing Internet? ☐ Wireless ☒ Wired ☐ Don't know

Where are you accessing Internet? ☐ Home ☐ Work ☒ School ☐ Public location ☐ Other ☐ Don't know

Please select applications and sites of interest to you:

News:

☐ CNN
☐ Los Angeles Times
☐ MSNBC
☐ New York Times
☐ Washington Post
☐ ABC News
☐ Reuters
☐ BBC
☐ Huffington Post
☐ USA Today

Social Networks:

☐ Facebook
☐ Twitter
☐ MySpace
☐ Friendster

☐ File Hosting Services (ex. Rapidshare, Hotfile, Depositfiles etc.)

Online Games:

☐ World of Warcraft
☐ League of Legends
☐ Call of Duty 2
☐ Call of Duty 4
☐ Team Fortress 2

Video Only Conference:

☐ US East Coast
☐ US West Coast
☐ Europe

Voice Conference:

Figure 6 – Users Feedback about Their Connection

As shown in Figure 7, users can mouseover rating to get an explanation of how a rating was determined.

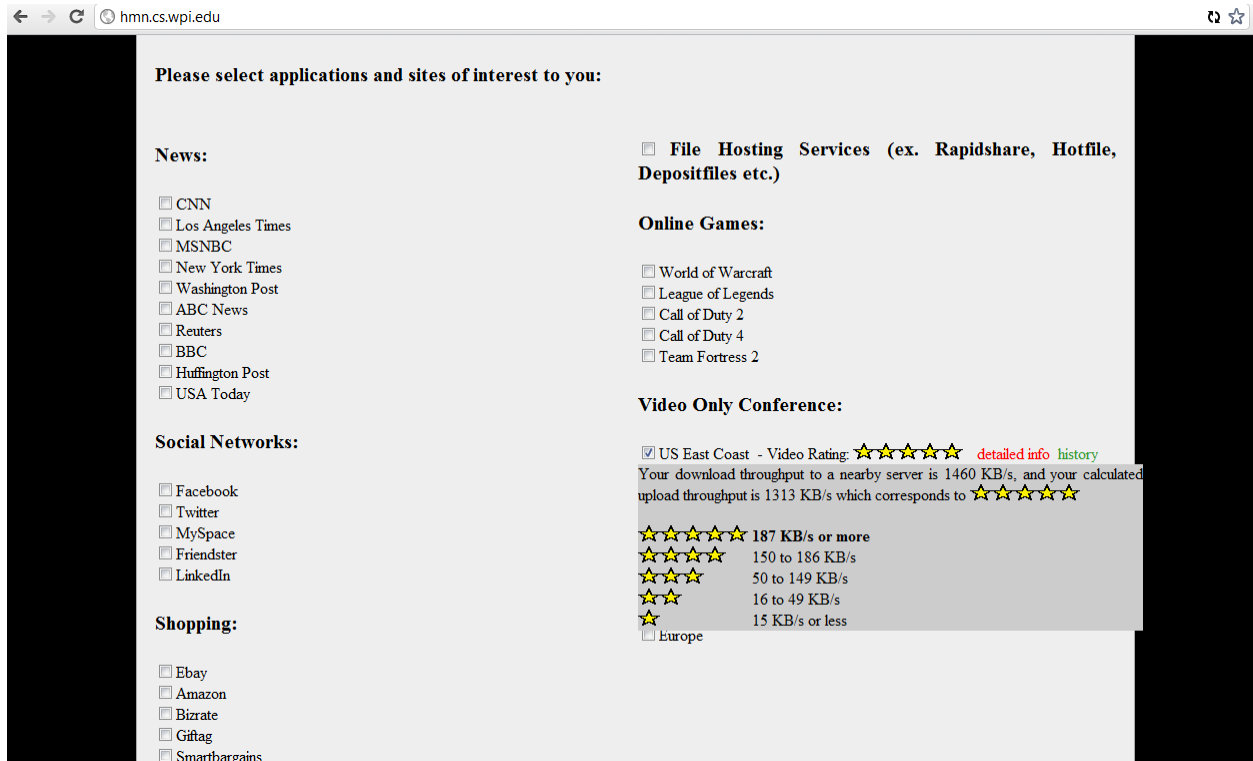


Figure 7 – Explanation about Ratings

If users are more interested in the measurements that contributed to their ratings, they can mouseover detailed info, as shown in Figure 8.

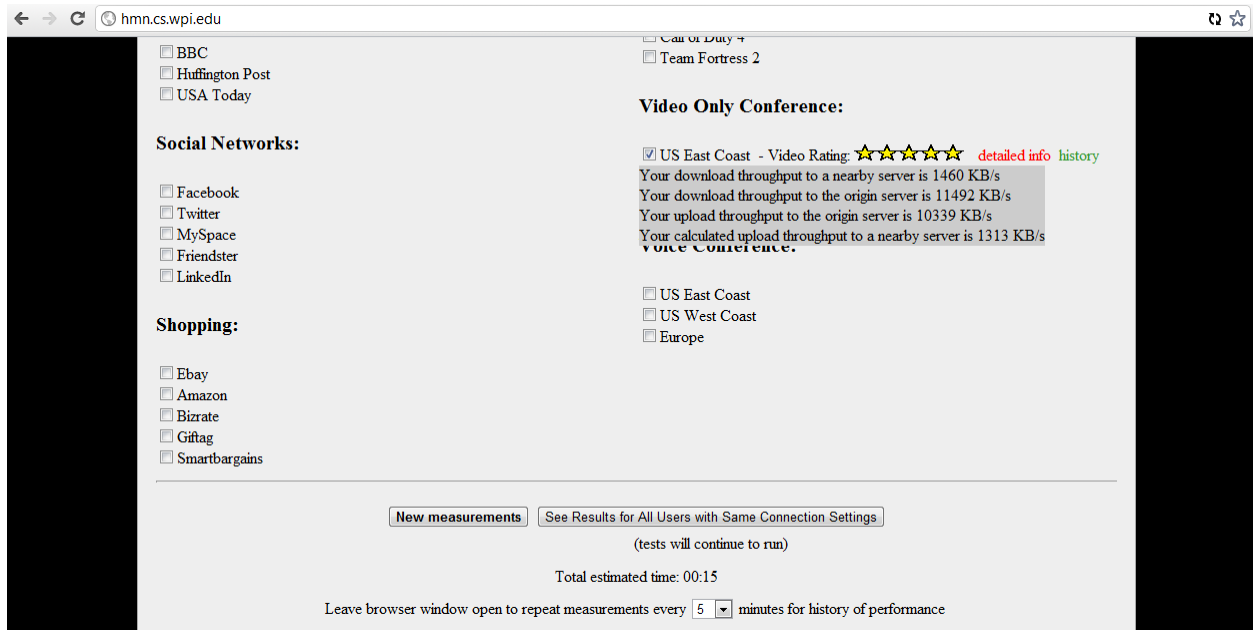


Figure 8 – Detailed Explanation about Ratings

Users are also provided with the possibility of tracking their ratings over time by doing mouseover history label, as shown in Figure 9.

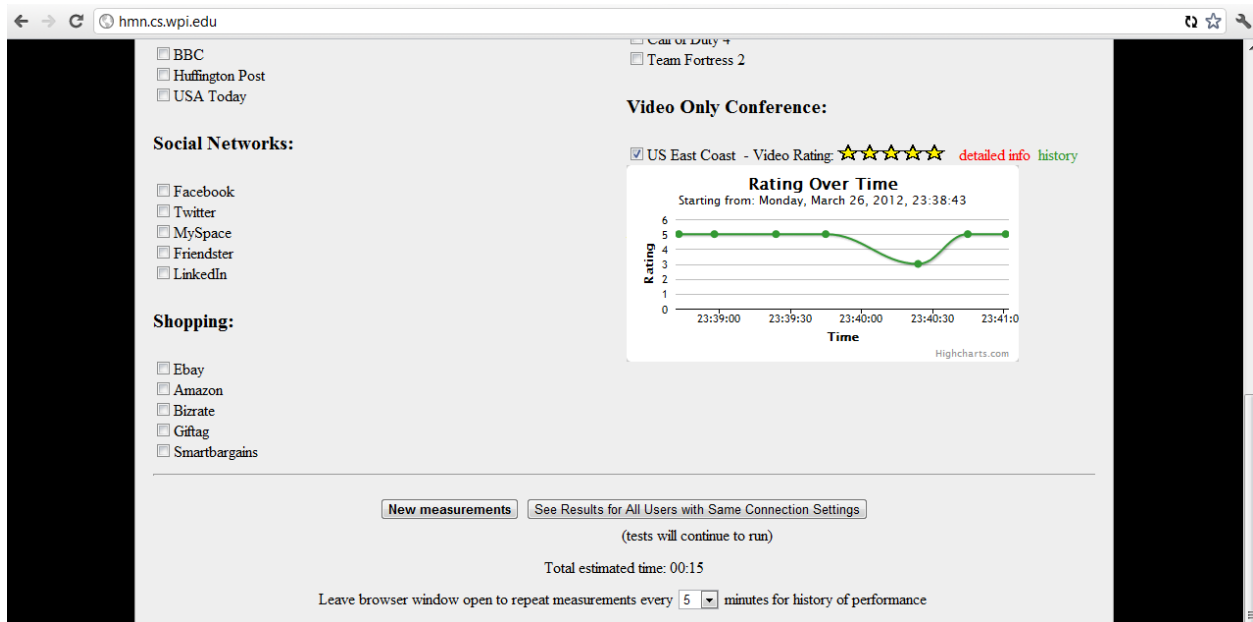


Figure 9 – Session History of Ratings

They can also give us feedback about the activities for which they would like us to measure performance in the future and anything else they would like to share, as shown in Figure 10.

The screenshot shows a web browser window with the address bar displaying 'hmn.cs.wpi.edu'. The page content is as follows:

- Top Left:** A list of social media links: MySpace, Friendster, and LinkedIn, each with an unchecked checkbox.
- Top Right:** A section titled 'Voice Conference:' with three unchecked checkboxes: US East Coast, US West Coast, and Europe.
- Shopping:** A section with the title 'Shopping:' and a list of e-commerce sites: Ebay, Amazon, Bizrate, Giftag, and Smartbargains, each with an unchecked checkbox.
- Buttons:** Two buttons are present: 'Perform All Measurements' and 'See Results for All Users with Same Connection Settings'.
- Text:** Below the buttons, it says '(tests will continue to run)' and 'Total estimated time: 00:00'.
- Form:** A text input field with the label 'Leave browser window open to repeat measurements every' followed by a dropdown menu set to '5' and the text 'minutes for history of performance'.
- Section:** A section titled 'In terms of future work, are you interested in the performance of any of the other applications below?'.
- List:** A list of applications with unchecked checkboxes: Music Streaming (ex. Pandora), Video Streaming (ex. YouTube), Live Video Streaming (ex. ESPN), Internet Radio (ex. BBC), Search Engines(ex. Google), P2P File Sharing (ex. BitTorrent), and Other (with a text input field).
- Text:** A line of text: 'Please provide written feedback about anything else you would like to share with us about the site:'.
- Form:** A large text input field for written feedback.
- Buttons:** Two buttons at the bottom: 'Clear' and 'Send List of Other Applications and Written Feedback'.
- Footer:** A footer section containing the text 'How's My Network | Worcester Polytechnic Institute | hmnn@cs.wpi.edu' and a disclaimer: 'This material is based upon work supported by the National Science Foundation under Grant No. 0959441. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.'

Figure 10 – Users Feedback about Additional Activities and Feedback They Want to Share

6.3 Database

Test and survey results are sent to the MySQL database using a Perl Script. All results are stored in one table that has a date, cookie, IP address, browser data, test results, survey results, automated testing and session test number.

Browser data, test results and survey results are represented as a hash table in JSON format. Browser data contains name and version of the browser and the operating system. Test results contain all low-level measurements we made and all calculated values that were presented to the user. Survey results contain information input by the user.

Automated testing represents whether the test was executed automatically or a user was the one who initiated it. Session test number represents the order of measurement for the current session. Every time a user opens the application, this number is reset.

Test results are sent every time a measurement for all selected activities is finished. User selection for location, connection and access type is sent along with every test. If a user wants to leave feedback about additional activities or a comment, the user can send that data separately from the measurements by clicking the appropriate button at any time.

All the information that comes from users is added to the database table using *insert* function in MySQL and is done quickly even if the table is large. Displaying information about other users activity ratings requires getting all entries from the table and it is therefore not done at run time. It is updated automatically on an hourly basis. Once every hour the server runs a Perl script that generates all information in a text file as a JSON object. This text file can be retrieved and parsed by users later.

6.4 Summary

We wanted to make a Web application that allows users to perform network measurements for online activities they are interested in and also provide feedback about it. We made an application where users can go to, select activities they are interested in and perform

measurements. Results are displayed on a five-star rating scale and users also have an ability to view more details about how these ratings were calculated.

7. Results

We had our first participants on March 16th 2012 and for this thesis we will analyze all results gathered until April 12th 2012.

7.1 User Stats

Total number of users that used our tool to perform measurements was 189. We identified users by cookie and IP address. 175 of them left some feedback about their connection.

Table 6 shows the number of users in different groups. Groups are made according to the geographical location, type of location and type of connection. Geographical location was determined using IP address and MaxMind Geo IP [16].

Table 6 – Number of Users in Different Groups

	Group	Number of Users
All	All	189
Geographical Location	US	182
	Massachusetts	161
	Worcester	98
	WPI	62
	non-WPI	127
Type of Location	Home	101
	Work	18
	School	46
	Public	3
	Other	4
Type of Connection	Dial-up	0
	Cable	77
	DSL	21
	Fiber	17
	Business	20
	Other	4

7.2. Low-Level Results

Table 7 shows the results for low-level measurements for users measuring in different locations. It includes number of users that belong to the group, total number of measurements performed by them, number of them that selected activities with throughput measurements and the values for the measurements. File hosting services and video only conference are the only two activities that require throughput measurements.

Throughput is first averaged per user and then averaged for all users. Users at WPI had the best performance for both download and upload throughput.

Table 7 – Low-Level Measurements Results for Different Locations

	Number of Users	Number of Measurements	Users who Selected Activities with Throughput Measurements	Users Download Throughput to a Nearby Akamai Server [KB/s]	Users Upload Throughput to a Nearby Akamai Server [KB/s]
All	189	824	103	2096	1469
US	182	816	99	2175	1501
Massachusetts	161	771	89	1951	1201
Worcester	98	371	50	2213	1721
WPI	62	180	31	2754	1964
non-WPI	127	644	72	1812	1253

Table 8 shows throughput results for users at different types of location. Users at work location had the best performance and users at public locations had the worst. Users at public locations also have significantly higher upload throughput compared to download. The reason for this could be a small number of users since only two of them performed throughput measurements.

Table 8 – Low-Level Measurements Results for Different Types of Location

	Number of Participants	Number of Measurements	Users who Selected Activities with Throughput Measurements	Users Download Throughput to a Nearby Akamai Server [KB/s]	Users Upload Throughput to a Nearby Akamai Server [KB/s]
All	189	824	103	2096	1469
Home	101	433	59	1412	696
Work	18	63	7	5738	6070
School	46	120	27	2885	2123
Public	3	5	2	146	354
Other	4	3	1	2720	470

Table 9 shows throughput results for users with different types of connection. We did not have any users with dial-up connection. Users with business connection had the best download and users at locations not on our list had the best upload throughput, although we only have one user with measurements in this group. Users with DSL connection had the worst performance. Cable and DSL users have about two times larger download than upload throughput. Other users have closer values for the two throughputs.

Table 9 – Low-Level Measurements Results for Different Types of Connection

	Number of Participants	Number of Measurements	Users who Selected Activities with Throughput Measurements	Users Download Throughput to a Nearby Akamai Server [KB/s]	Users Upload Throughput to a Nearby Akamai Server [KB/s]
All	189	824	103	2096	1469
Dial-up	0	0	-	-	-
Cable	77	204	46	1520	765
DSL	21	69	12	833	277
Fiber	17	205	9	2720	2133
Business	20	85	10	6215	5197
Other	4	4	1	5868	5294

Table 10 shows throughput results for users with different types of Internet Access. Users with wired connection performed a lot better performance than the users with wireless.

Table 10 – Low-Level Measurements Results for Different Types of Internet Access

	Number of Participants	Number of Measurements	Users who Selected Activities with Throughput Measurements	Users Download Throughput to a Nearby Akamai Server [KB/s]	Users Upload Throughput to a Nearby Akamai Server [KB/s]
All	189	824	103	2096	1469
Wireless	103	569	63	1084	768
Wired	68	244	36	3908	2761

7.3 Activities Popularity

Table 11 contains list of all activities a user can select on our site along with the percent of them that selected each. Facebook and Amazon were two most popular activities and they were selected by 72 and 74 percent of the users respectively. CNN and NY times are two most popular news sites selected by 38 and 32 percent of the users respectively. File hosting services were selected by 41 percent of the users. Most popular first person game was Team Fortress 2. World of Warcraft and League of Legends on US servers were selected by 15 and 14 percent of our users respectively. About one third of users selected voice and video conference to the US east coast.

Table 11 – Percent of Users that Selected Each of the Activities

	Activity	Percent
News	CNN	38
	LA	9
	MSNBC	12
	NY Times	32
	Washington Post	9
	ABC News	13
	Reuters	15
	BBC	29
	Huffington Post	15
	USA Today	14
Social Networks	Facebook	72
	Twitter	17
	My Space	5
	Friendster	4
	LinkedIn	29
Shopping	Ebay	35
	Amazon	74
	Bizrate	5
	Giftag	4
	Smart-bargains	4
File Hosting	File Hosting	41
WoW	WoW US	15
	WoW Australia	3
	WoW Europe	4
	WoW China	5
	WoW Taiwan	3
	WoW Korea	3
LoL	LoL US	14
	LoL Europe	2
	LoL Asia	2
First Person Games	Call of Duty 2	6
	Call of Duty 4	11
	Team Fortress 2	18
Video Conference	Video Conference US East Coast	35
	Video Conference US West Coast	18
	Video Conference Europe	15
Voice Conference	Voice Conference US East Coast	33
	Voice Conference US West Coast	16
	Voice Conference Europe	15

7.4 User Activity Performance

In Figure 11 we can see percent of users that had each of the ratings for all news Web sites. For calculating number of users, we took measurements for each user and put the fraction of each rating value received in that bin. For example if for a certain activity a user gets five stars 10 times and four stars 10 times, we count 0.5 users for five stars and 0.5 users for four stars.

Most of the users got five stars for each of the news except Los Angeles Times. This Web site also had the most users with one star rating. The reason for this could be that most of our users are located on the US east coast.

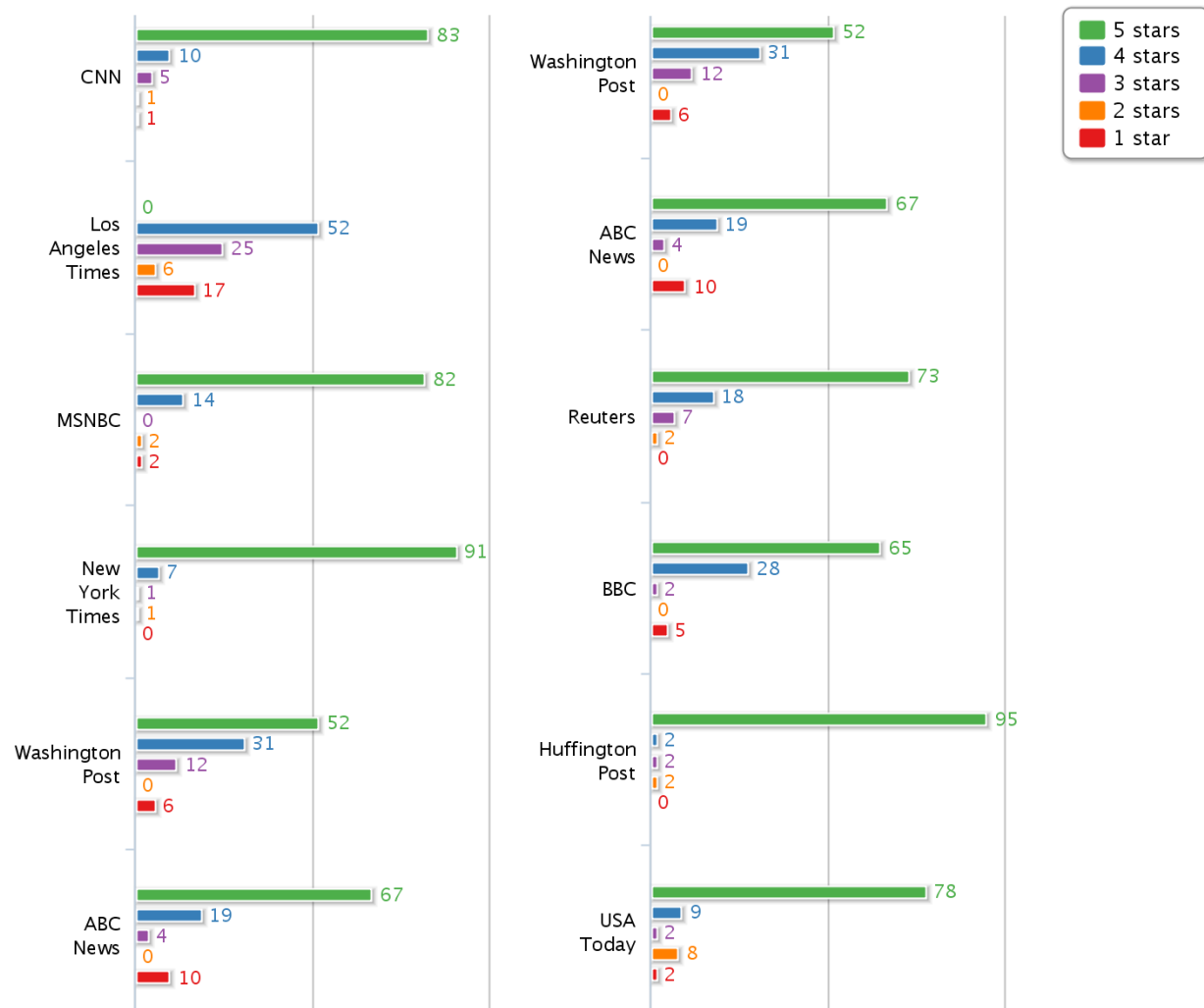


Figure 11 – Rating Percentages for News Web Sites

Figure 12 shows rating results for social networks and shopping sites. All social networks and almost all shopping sites had great majority of users with five star ratings. Smartbargains only had 17 percent of users with five stars.

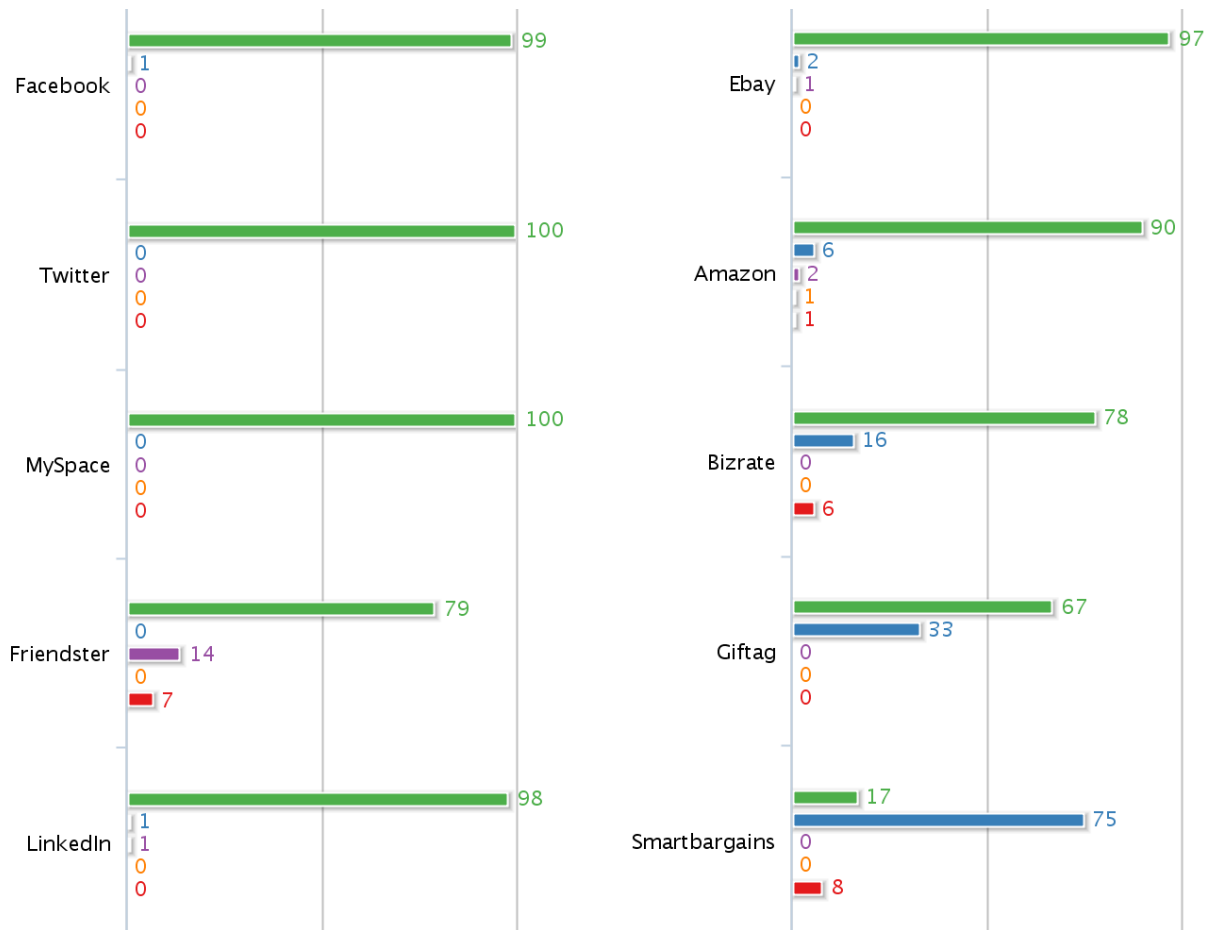


Figure 12 – Rating Percentage for Social Network and Shopping Sites

In Figure 13, we can see the results for file hosting services. About one third of the users had five stars and about one third of them had four stars.

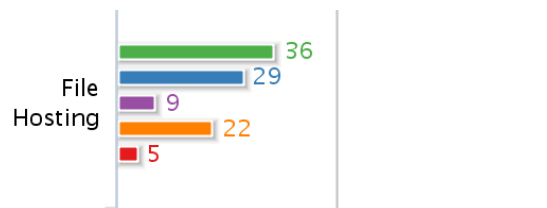


Figure 13 – Rating Percentage for File Hosting Services

Figure 14 shows users results for playing online games. League of Legends had almost all users with five star ratings for all servers. This makes sense because it is omnipresent game and as therefore has better delay tolerance.

World of Warcraft had 90 percent of users with five stars for US servers which is the closest one for most of them. For servers in Europe which is a bit further, only 71 percent of them managed to get five stars. For servers in Asia which is a lot further only between 20 and 25 percent of users managed to get five stars.

First person games only had about one third of users with five stars and one third with four stars. This makes sense because they are the least delay tolerant and require less than 26 milliseconds for five star rating.

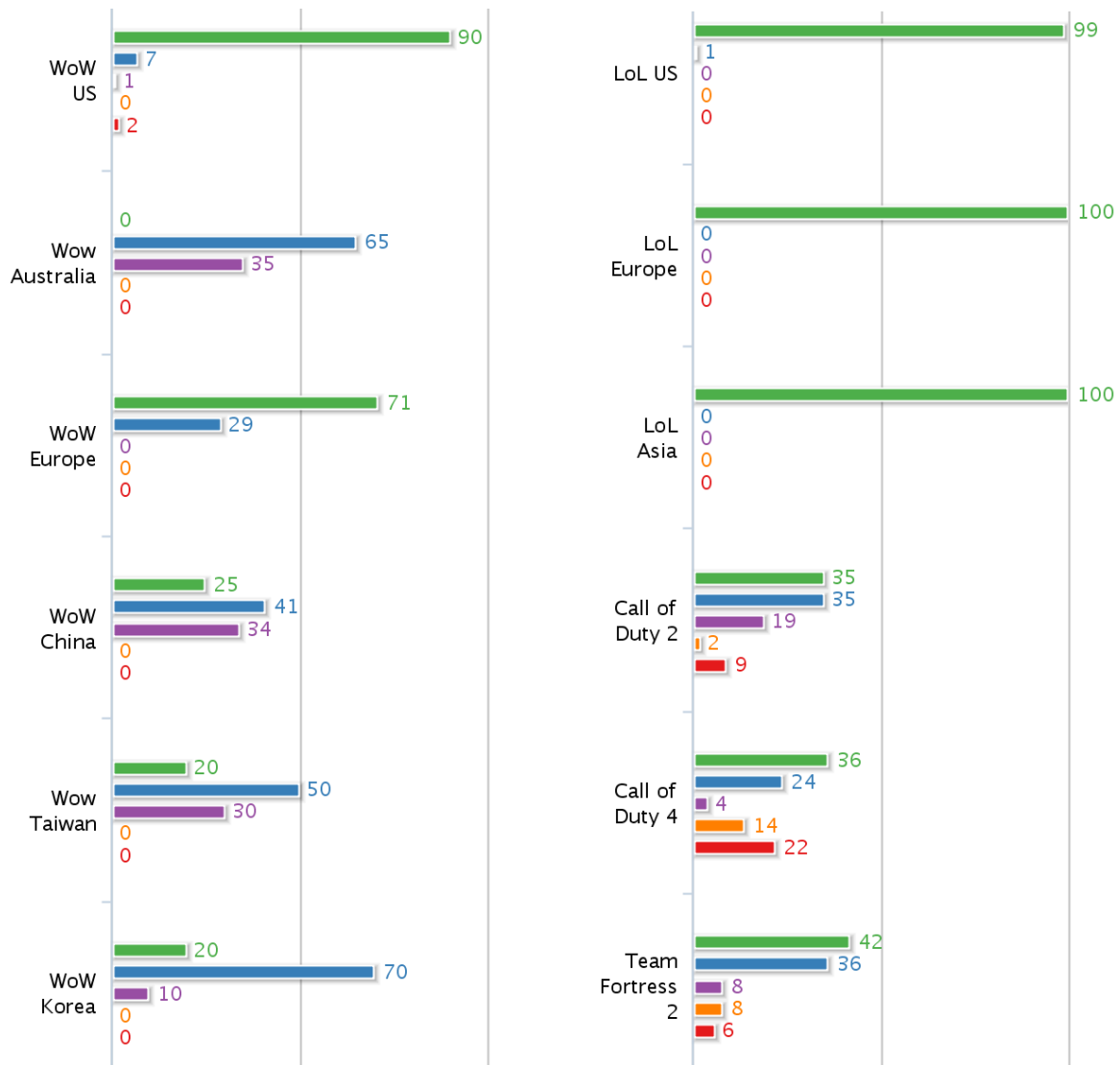


Figure 14 – Rating Percentage for Online Games

In Figure 15 we can see the rating results for voice and video conference. Most of the users got five stars for each activity. The number is a bit smaller as we go away from US east coast and the reason for this could also be that most of our users are located in this area.

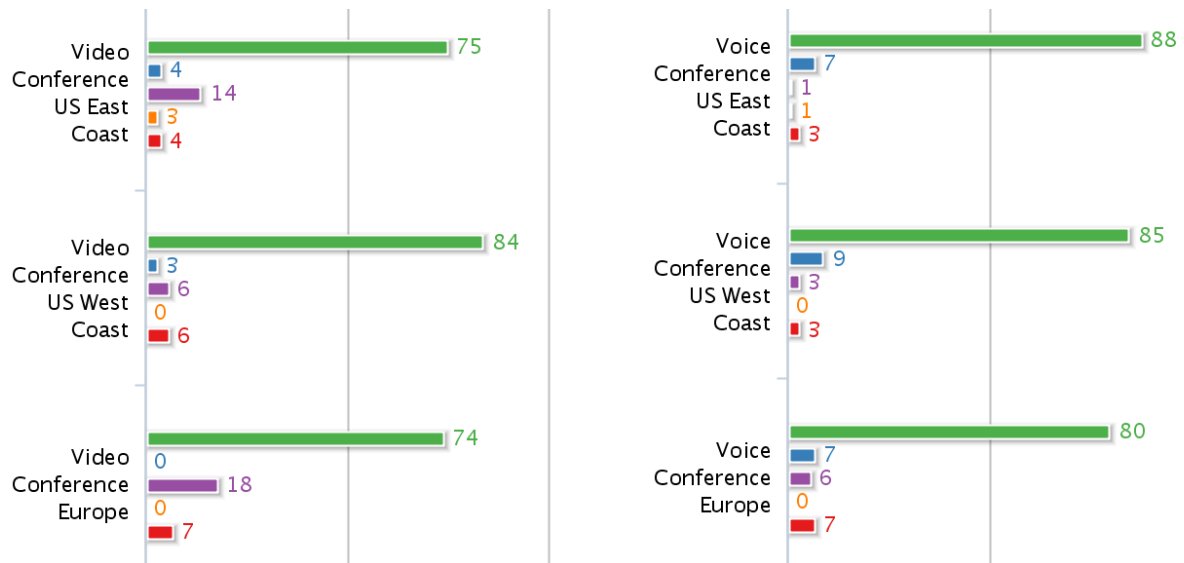


Figure 15 – Rating Percentage for Voice and Video Conference

In Figure 17 we have comparison of ratings for users at home and users at work and school location for social networks and online shopping sites. Home users have about 25 percent more five star ratings for Washington Post, Reuters and BBC. They also have about 20 percent less users with five star ratings for MSNBC.

None of the work or school users had one star rating for any activity. Home users had 25 percent of these users for Freindster, Bizrate and Smartbargains. They also had about 40 percent less five star users for Giftag.



Figure 16 – News Sites Rating Comparison for Home and Work/School Users

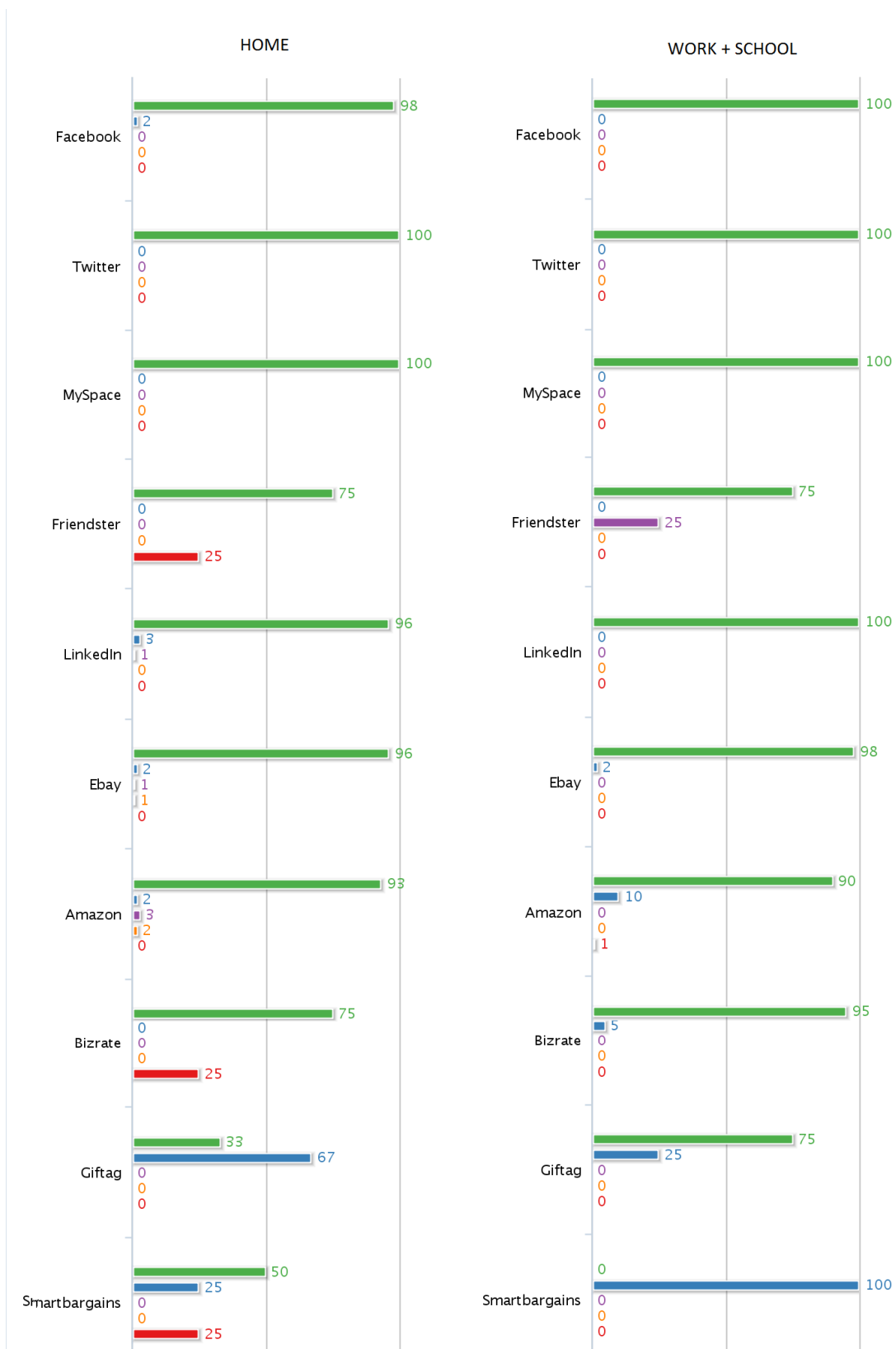


Figure 17 – Social Network and Online Shopping Sites Rating Comparison for Home and Work/School Users

In Figure 18 we have comparison of ratings for users at home and users at work and school location for file hosting services. We have a lot more users with five stars that are in work or school and a lot less of them with four stars.

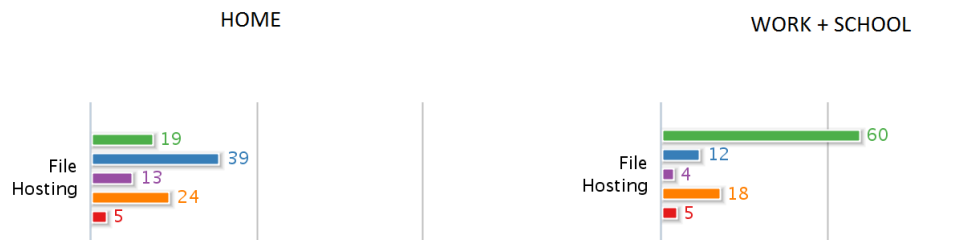


Figure 18 – File Hosting Rating Comparison for Home and Work/School Users

Figure 19 shows the comparison of ratings for World of Warcraft and League of Legends for which we have data in both groups. There are no significant differences between the groups.

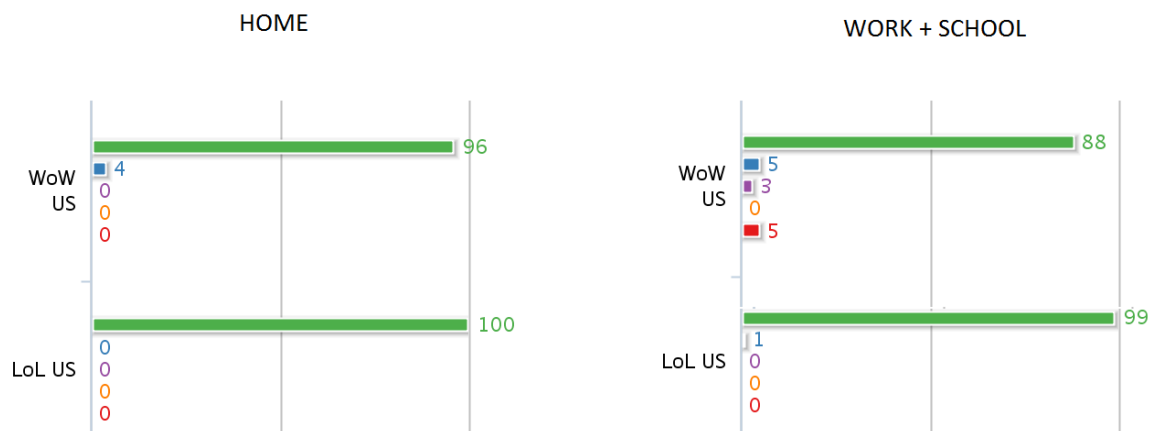


Figure 19 – WoW and LoL Rating Comparison for Home and Work/School Users

Figure 20 shows the comparison of ratings for first person games. Users at work or school have a lot better performance for all. They have more users with five stars and less with one star.

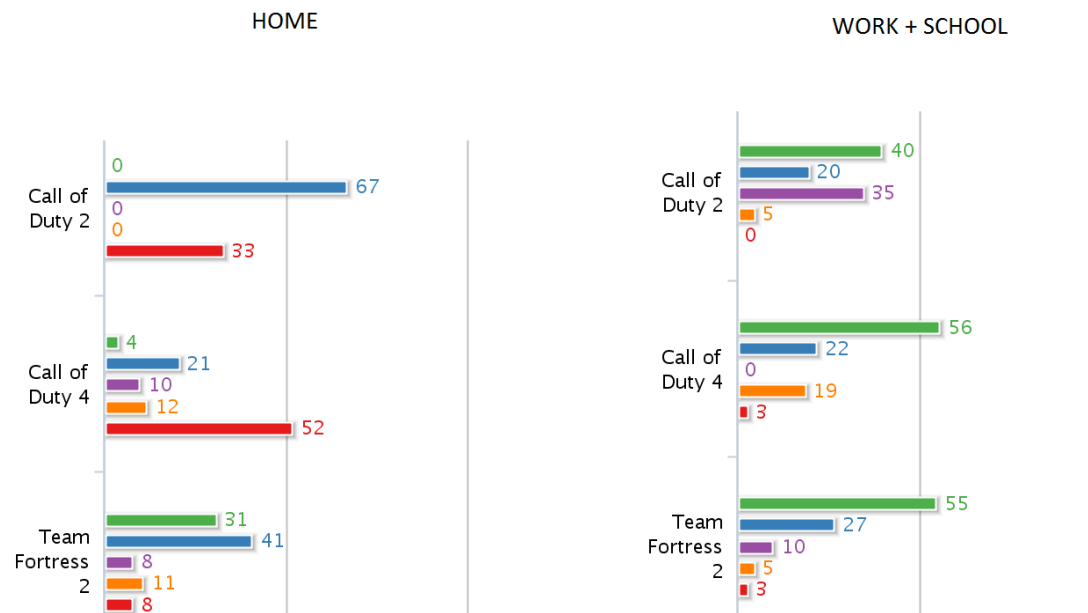


Figure 20 – First Person Games Rating Comparison for Home and Work/School Users

Figure 21 shows the comparison of ratings for voice and video only conference. Home users performed a lot better for video only conferencing. They have about 20 percent more users with five stars and no user with one star. Work and school users have about 7 to 9 percent of users with one star for video only conferencing.

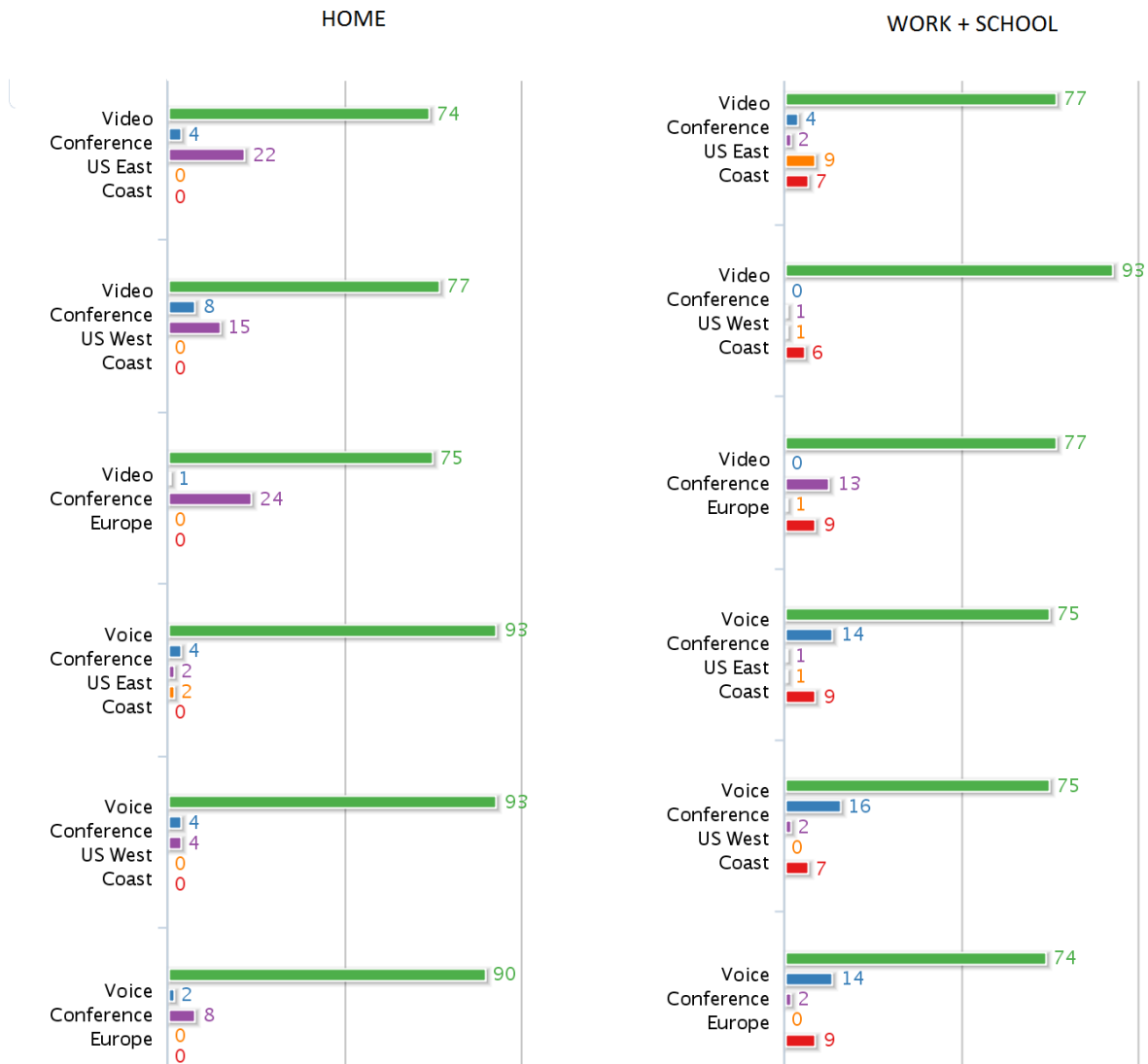


Figure 21 – Voice and Video Only Conference Rating Comparison for Home and Work/School Users

7.5 Feedback and Additional Activities

Five of our users left a written feedback such as:

- “Speed stats, animations such as speedtest.net perhaps”
- “Cool Site! Nice Job”

We had 52 users answering which additional activities they would like us to add to our Web application. Table 12 contains percent of those users that selected each of the additional activities.

Table 12 – Percent of Users that Selected Each of the Additional Activities

Activity	Percent of Users that Selected the Activity
Video Streaming	93
Search Engines	77
Music Streaming	74
Live Video Streaming	51
P2P File Sharing	47
Internet Radio	43

From the activities that were not on the list users wanted to see the following:

- Large ftp/http download from kernel.org
- foxnews.com
- Source code transfer

8. Packet Pair

Packet pair technique is used to measure bottleneck link capacity [24]. This is another network measurement of interest to us and we want to see how accurately we can get it within a Web browser.

8.1 Calculations

Two packets of the same size traveling from the same source to the same destination have a time difference of:

$$t_2^d - t_1^d = \max\left(\frac{s}{b}, t_1^s - t_2^s\right)$$

where t_1^d and t_2^d are destination arrival times of first and second packet respectively, t_1^s and t_2^s are source departure times of first and second packet respectively, s is size of packets, and b is the bottleneck link bandwidth.

8.2 Assumptions

For packet-pair technique to give meaningful results, we need to make some assumptions:

1. Queue at the Bottleneck Only

Two packets have to be queued together at the bottleneck and no later link.

2. Packets Grouped Together

There are no other packets in queued between them or in front of them.

3. FIFO queue

Bottleneck router is using FIFO queuing.

4. Transmission Delay

Transmission delay is proportional to packet size and routers are store-and-forward

8.3 Measurements

We can implement packet pair using JavaScript DOM and XHR techniques. We download six files at the same time and measure the difference between receiving consecutive responses. Since all requests are executed at the same time, the formula above can be simplified to:

$$t_2^d - t_1^d = \frac{s}{b}$$

We can calculate the bottleneck bandwidth from here:

$$b = \frac{s}{t_2^d - t_1^d}$$

We perform download of six packets 4 times for Chrome, Firefox and Internet Explorer running in Windows 7. For each of the measurements we also record a difference between received packets from WireShark [25].

For size of the packet we used size of the file downloaded plus HTTP, TCP, and IP header sizes. We disregard Ethernet header because it is attached by router at the end of the path.

Figure 22 shows a CDF graph of measurements we performed to the NY Times server using DOM technique. On the x-axis we have the calculated bottleneck link bandwidth and on the y-axis we have percent of measurements that are above this value.

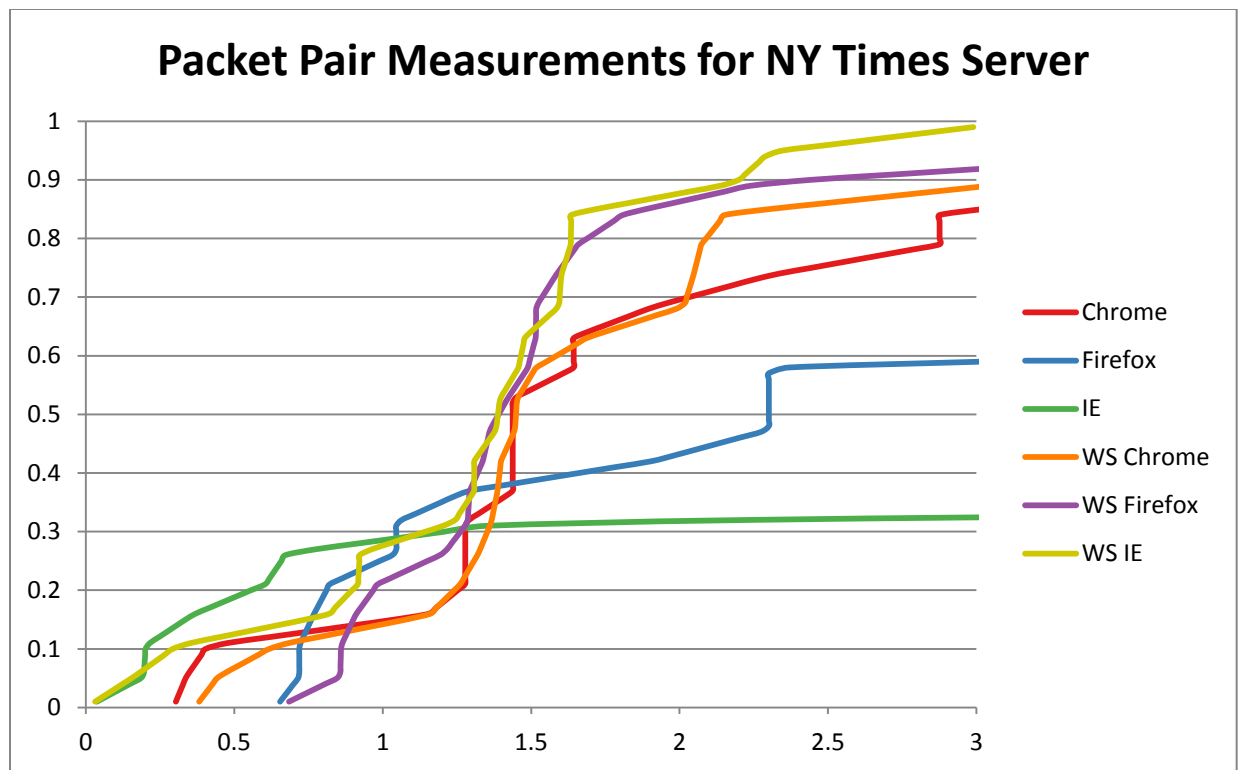


Figure 22 – CDF Graph for Packet Pair Measurements for NY Times Server from a Public Location

If the Web browser had no overhead, measurements from the application and WireShark would be the same. We can see from the graph that Chrome is close to its corresponding WireShark measurements and medians are almost the same. Firefox and Internet Explorer results do not closely match the corresponding WireShark results.

Figure 23 shows a CDF graph of measurements we performed to the BBC server.

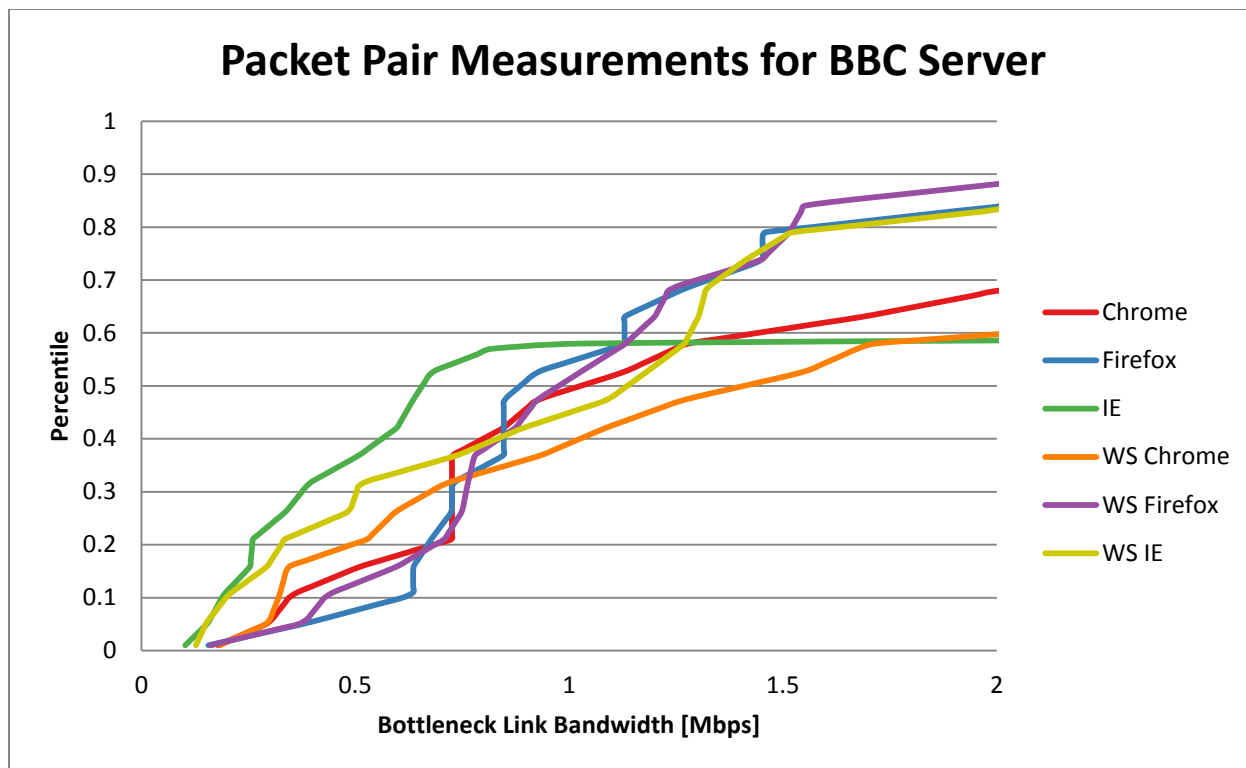


Figure 23 – CDF Graph for Packet Pair Measurements for BBC Server from a Public Location

Chrome and Firefox measurements are close to the corresponding ones obtained by WireShark. Medians for Firefox are close and medians for Chrome are less than 0.5 Mbps apart. Internet Explorer measurements do not closely follow corresponding Wireshark measurements.

Figure 24 shows a CDF graph of measurements we performed to the Facebook server.

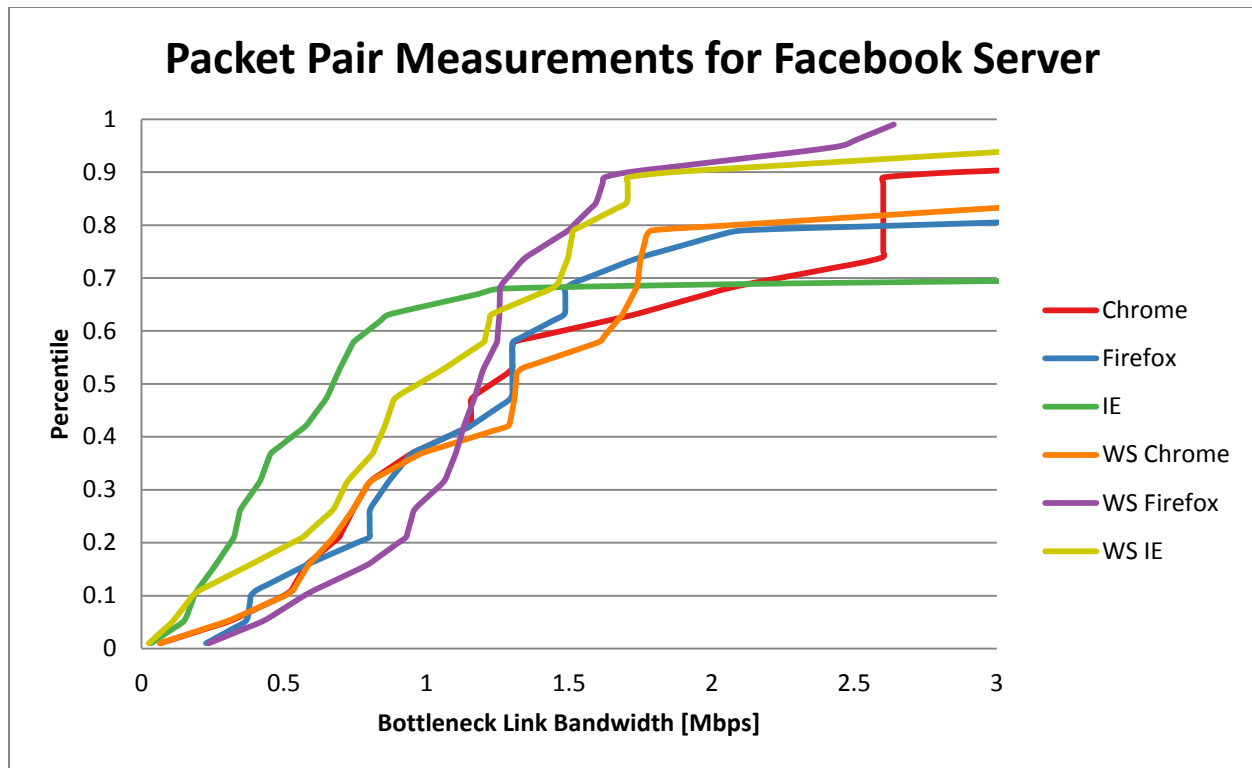


Figure 24 – CDF Graph for Packet Pair Measurements for Facebook Server from a Public Location

Chrome and Firefox measurements are close to the corresponding ones obtained by WireShark and their medians are close. Internet Explorer measurements do not closely follow corresponding Wireshark measurements, but their medians are less than 0.5 Mbps apart.

Figure 25 shows a CDF graph of measurements we performed to the Ebay server.

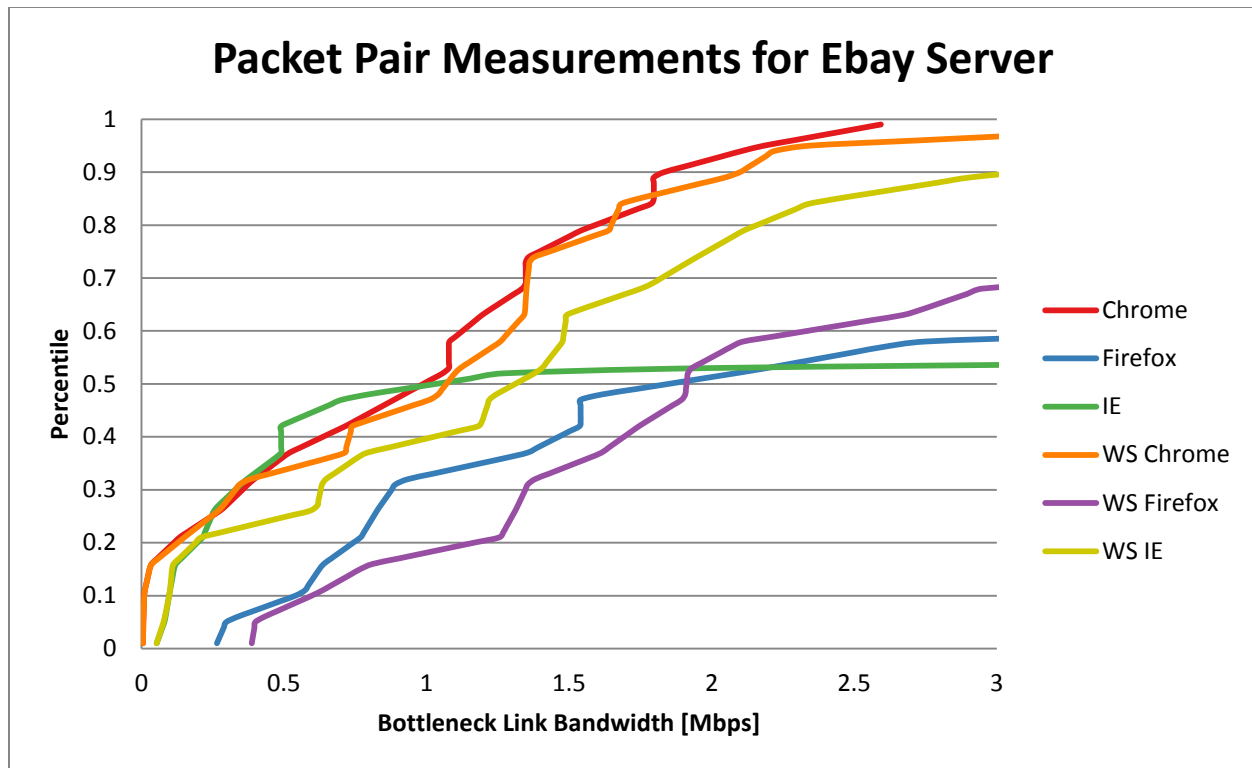


Figure 25 – CDF Graph for Packet Pair Measurements for Ebay Server from a Public Location

Chrome and Firefox measurements are close to the corresponding ones obtained by WireShark and their medians are almost the same. Internet Explorer measurements do not closely follow corresponding Wireshark measurements, but their medians are close.

Figure 26 shows a CDF graph of measurements we performed to the origin server using XHR technique. We also used IGI/PTR [26] program for measuring bottleneck link available bandwidth which is capacity decreased by used bandwidth. It calculates the bandwidth using two different methods: IGI and PTR. PTR is a method they recommend and in the graph we show the result of 20 measurements we had.

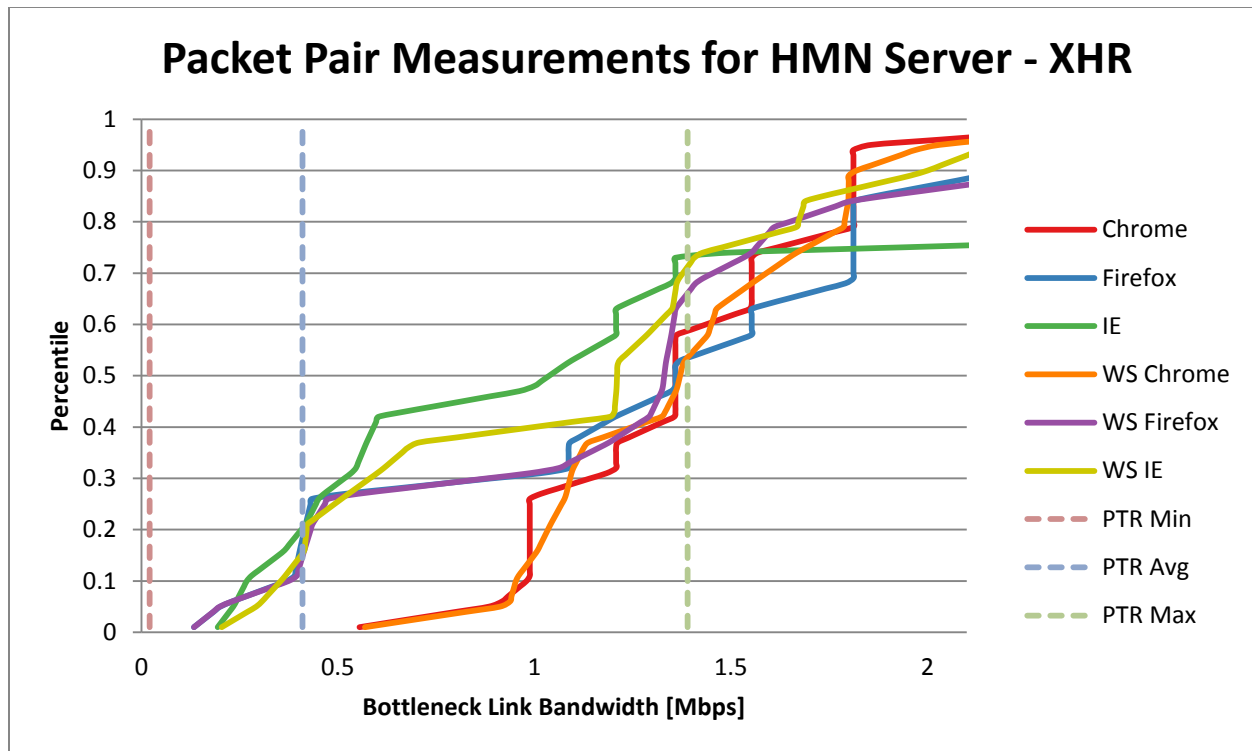


Figure 26 – CDF Graph for Packet Pair Measurements for Origin Server from a Public Location

All three browser measurements are close to the corresponding ones obtained by WireShark. Medians for Chrome and Firefox are almost the same and Internet Explorer medians are close. Minimum and average bandwidth calculated by IGI/PTR are significantly smaller than any of the WireShark medians. Maximum however is close to these values.

8.4 Web Site Results

Currently in our Web application we measure time differences between two packets from CNN Web site five times along with other measurements using DOM method. This is the early approach we had for doing packet pair measurements using Web browser. Measurements we did later showed that downloading six objects instead of two produces time differences in the Web Browser that are a lot closer to the results obtained by WireShark.

Table 13 shows the summary of the results for users at different locations. For each measurement we took the median of five, averaged per user and took the median of all users.

Table 13 – Packet Pair Results for the Web Application Users at Different Locations

Location	Number of Participants	Packet Pair Bottleneck Bandwidth Median [Mbps]
All	186	0.73
US	179	0.78
Massachusetts	159	0.86
Worcester	96	0.76
WPI	61	0.73
non-WPI	125	0.76

All the values we obtained for bandwidth are between 0.73 and 0.86. Table 14 shows packet pair results for users with different types of location.

Table 14 – Packet Pair Results for the Web Application Users at Different Types of Locations

Location	Number of Participants	Packet Pair Bottleneck Bandwidth Median [Mbps]
All	186	0.65
Home	101	0.94
Work	18	0.20
School	46	0.46
Public	3	0.14
Other	4	1.61

Users that selected to be at location other than home, work or school had the best performance, but the number of users we got information from is small. Users at public locations performed worst. Users at work performed worse than users at home. This is surprising considering the throughput results we had before and we should consider reexamining how we currently do measurements on the site.

Table 15 shows packet pair results for users with different types of Internet connection.

Table 15 – Packet Pair Results for the Web Application Users with Different Types of Connection

Location	Number of Participants	Packet Pair Bottleneck Bandwidth Median [Mbps]
All	186	0.65
Dial-up	0	-
Cable	77	1.08
DSL	21	0.37
Fiber	17	0.57
Business	20	0.17
Other	4	0.47

Users with cable connection performed best and users with business connection performed worst. These results are surprising considering the throughput measurements we had before and they also confirm we should do the reexamination of the packet pair methodology that is currently implemented.

Table 16 shows packet pair results for users with different types of Internet access.

Table 16 – Packet Pair Results for the Web Application Users using Different Types of Internet Access

Location	Number of Participants	Packet Pair Bottleneck Bandwidth Median [Mbps]
All	186	0.65
Wireless	103	0.61
Wired	68	0.66

Wired users performed a bit better than wireless users.

8.5 Summary

Packet-pair is a technique we can use to measure bottleneck bandwidth. It includes sending two or more packets from source to destination at the same time and measuring the gap at the destination. We can calculate bottleneck bandwidth from this gap and packet sizes. We tested this technique in Web browsers by downloading 6 packets at the same time. We also

implemented this technique in our Web application by loading two files concurrently using DOM method and measuring the time difference between onload events. Results show significant difference between different types of locations and different types of connections. Geographical location and type of access did not have much effect on packet pair results.

9. Conclusions

From the results of our work, we can conclude that Web browsers can be used reliably for RTT and download and upload throughput measurements. These low-level measurements can be used to estimate performance of many online activities such as browsing different categories of websites, having voice or video conferencing, playing online games and using file-hosting services.

Minimal impediments and good incentives can help us gather data from a lot of users. Making the tool more available to users, we increase the chance that they are willing to use it.

10. Future Work

We want to continue expanding set of participants. Currently we have 189 of them. Some of them gave us suggestions about what activities they would like us to implement on our website. We can use these as guidelines and also expand the set of activities we currently offer.

In this work we used some of the methods that are still not evaluated. We compensate for Web browser overhead by deducting the time it takes a user to download an object from cache. For loss rate evaluation, we take percent of RTTs that are at least three times the median. Another assumption we made without a good validation is considering users download/upload ratios to be the same to any server.

We should also benchmark our activity performance estimate with the actual performance. This has been done only for Web site activities, but we would like to know how good our predictions are for the others too.

One of the considerations for future work would also be building a similar tool for mobile devices. This will give us better insight into the performance of mobile users.

11. References

- [1] What Internet Users Do Online | Pew Research Center's Internet & American Life Project, December 2010, Pew Research Center, <http://www.pewinternet.org/Trend-Data/Online-Activites-Total.aspx>
- [2] Network Dashboard, Georgia Tech, <http://projectbismark.net/>
- [3] NETI@home: A distributed approach to collecting end-to-end network performance measurements Charles Robert Simpson, Jr. and George F. Riley, In PAM2004 - A workshop on Passive and Active Measurements, April 2004.
- [4] The DIMES project. <http://www.netdimes.org/>.
- [5] DipZoom: The Internet Measurements Marketplace, Michael Rabinovich, Sipat Triukose, Zhihua Wen, and Limin Wang. In Proceedings of the 9th IEEE Global Internet Symposium, May 2006.
- [6] Understanding End-user perception of Network Problems, J. Scott Miller, Amit Mondal, Rahul Potharaju, Peter A. Dinda, Aleksandar Kuzmanovic, In Proceedings of ACM SIGCOMM Workshop on Measurements Up the STack (W-MUST), Toronto, CA, August 2011.
- [7] Netalyzr: Illuminating The Edge Network, Christian Kreibich, Boris Nechaev, Vern Paxson, Nicholas Weaver, ACM IMC 2010.
- [8] How's my network? A Java approach to home network measurement. Alan Ritacco, Craig E. Wills, and Mark Claypool, In Proceedings of the IEEE International Conference on Computer Communications and Networks, San Francisco, CA USA, August 2009.

- [9] Speedtest.net – The Global Broadband Speed Test, www.speedtest.net
- [10] Network performance evaluation in a web browser, Artur Janc, Craig E. Wills, and Mark Claypool, In Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems, pages 370-377, Cambridge, MA USA, November 2009.
- [11] M. Zeljkovic, M. Kaplan, C. Wills, and M. Claypool. JavaScript and Flash Overhead in the Web Browser Sandbox. Technical Report WPI-CS-TR-10-14, Computer Science Department, Worcester Polytechnic Institute, 2010.
- [12] A credit-based home access point (CHAP) to improve application performance on IEEE 802.11 networks, Choong-Soo Lee , Mark Claypool , Robert Kinicki, Proceedings of the first annual ACM SIGMM conference on Multimedia systems, February 22-23, 2010, Phoenix, Arizona, USA
- [13] The E-model, a computational model for use in transmission planning, ITU-T Recommendation G.107, May 2000. ITU-T
- [14] Help for Skype: How much bandwidth does Skype need?,
<https://support.skype.com/en-us/faq/FA1417/How-much-bandwidth-does-Skype-need>
- [15] Game Tracker - Server Stats, List Search, Ranking, Banners, & Social Network for Gamers, www.gametracker.com
- [16] Geolocation and Online Fraud Prevention from MaxMind, www.maxmind.com
- [17] Latency Can Kill: Precision and Deadline in Online Games, Mark Claypool and Kajal Claypool, In Proceedings of the First ACM Multimedia Systems Conference (MMSys), (Invited paper), Scottsdale, Arizona, USA, February 2010.

- [18] MaxMind - GeoLite City Accuracy for Selected Countries,
http://www.maxmind.com/app/geolite_city_accuracy
- [19] Akamai: The Leader in Web Application Acceleration and Performance Management, Streaming Media Services and Content Deliver, <http://www.akamai.com/>
- [20] Kaplan, M. 2011, Predicting Performance for Reading News Online from within a Web Browser Sandbox, Masters Thesis, Worcester Polytechnic Institute, Worcester MA
- [21] How's My Network? Predicting Performance within a Web Browser Sandbox, Murad Kaplan, Mihajlo Zeljkovic, Mark Claypool and Craig Wills, Submitted for publication, April 2012
- [22] J. Nielsen. Website Response Times, June 2009. <http://www.useit.com/alertbox/response-times.html>
- [23] Peter J. Sevcik. Understanding How Users View Application Performance, July 2002.
<http://www.netforecast.com/Articles/BCRC22PerformanceZones.pdf>.
- [24] Packet-Pair Technique,
http://static.usenix.org/event/usits01/full_papers/lai/lai_html/node2.html
- [25] Wireshark - Go deep, <http://www.wireshark.org/>
- [26] IGI/PTR, <http://www.cs.cmu.edu/~hnn/igi/>